M-62-66-2
APRIL 1966

SECOND QUARTERLY REPORT

PCM TELEMETRY DATA COMPRESSION STUDY, PHASE II

(25 NOVEMBER 1965 TO 25 FEBRUARY 1966)

by
W. R. BECHTOLD
T. BJORN, JR.
J. E. MEDLIN

Contract No. NAS 5-9729

Prepared by
LOCKHEED MISSILES & SPACE COMPANY
Sunnyvale, California

for

GODDARD SPACE FLIGHT CENTER
Greenbelt, Maryland

FOREWORD

This report describes the work accomplished during the second quarter of Phase II of the PCM Telemetry Data Compression Study. The work was performed for the National Aeronautics and Space Administration under Contract No. NAS 5-9729 during the period between 25 November 1965 and 25 February 1966. The first quarter of Phase II is reported in Ref. 1. Phase I of the study, completed on 15 August 1965, is reported in Ref. 2.

PRECEDING PAGE BLANK NOT FILMED.

PRECEDING PAGE BLANK NOT FILMED.

SUMMARY

The objective of this study is to investigate ways to improve the utilization efficiency of NASA/GSFC PCM telemetry links through data compression. The study is currently in its second phase. The first phase included a literature search and subsequent evaluation of known data compression techniques and a study of time-and-channel identification.

Phase II extends the data compression technique evaluation of Phase I, investigates transmission link noise effects, analyzes the impact of compressed data on computer operations, and performs a conceptual design of a ground-based data compression and reconstruction system. Phase II is divided into the following tasks, beginning with Task VII:

- Task VII, S-49 Compression Model Analysis
- Task VIII, New Data Compression Model Evaluation
- Task IX, Theoretical Buffer Analysis
- Task X, Data Compression Mechanization Study
- Task XI, Theoretical Transmission Error Analysis
- Task XII, Real Time Computer Analysis
- Task XIII, Data Processing Computer Analysis

This report discusses the work performed during the second three months of Phase II and presents the program of activity for the next three months. The second quarter's activity was confined to Tasks VII through X.

TASK VII, S-49 COMPRESSION MODEL ANALYSIS

Task VII involves an extension of the compression technique evaluation by computer simulation under Phase I, using telemetry data from the S-49, OGO-A satellite.

Activity on this task during the second quarter of Phase II consisted of an exploratory multisensor compression and buffer simulation run on 184,000 input data samples. A sample compression ratio of 13.4 was obtained for the run, using the first-order disjoined interpolator. The run provided an excellent indication of buffer queue behavior in the absence of control; with the ratio of average input rate to readout rate equal to 0.98, the maximum queue length was 505 samples.

TASK VIII, NEW DATA COMPRESSION MODEL EVALUATION

Task VIII evaluates, by computer simulation, new data compression techniques which were developed late in the first phase of the study. These techniques include sample selectors as well as promising methods of buffer feedback for the purpose of queuing control. Computer programs were written and debugged for four new or modified sample selection techniques during the reporting period. They are: (1) the exponential interpolation, (2) short-term frequency component selection, (3) bit plane encoding, and (4) broadside data compression. In addition, a new queuing control test model designed to evaluate adaptive filtering and sample deletion with Goddard Space Flight Center telemetry data was programmed and compiled. The model operates within the framework of the simulated multisensor compression and reconstruction systems which were programmed during Phase I.

TASK IX, THEORETICAL BUFFER ANALYSIS

One of the results of the theoretical study of buffer queuing control (Ref. 1) was that a combination of the queue length and its time integral promises to be a particularly effective parameter to control the data tolerance. During the second quarter, a control system using this parameter was programmed as the first step in its computer evaluation.

TASK X, DATA COMPRESSION MECHANIZATION STUDY

During the second quarter, as in the first, all activity in Task X was concentrated on the transmission link simulation subtask. This subtask investigates, by computer simulation, the effect of ground transmission link noise on compressed data. During the reporting period, all simulation programs were completed, and production test runs were begun. The method and range of Pareto noise parameter variation was also established.

CONTENTS

PRECEDING PAGE BLANK

ILLUSTRATIONS

TABLES

Section 1
INTRODUCTION

The objective of this study is to investigate ways to improve the utilization efficiency of NASA/GSFC PCM telemetry links through data compression. The work is in three general areas:

- Study and evaluate data compression system techniques
- Study the effect of transmission link noise on compressed data
- Determine the requirements of a ground-based data compression and reconstruction system

Activity during Phase I was confined primarily to the first area. This included a literature search and subsequent evaluation of known data compression techniques and a study of time-and-channel identification (Ref. 1). The technique evaluation was done with the aid of digital computer simulation using, as a model, PCM telemetry data from the S-6, Explorer XVII satellite.

The study is currently in its second phase. Phase II extends the first-phase evaluation by using additional telemetry data from the S-49, OGO-A satellite and by investigating transmission link noise effects. Near the end of the second phase, a study will be made of the configuration of a data compression and reconstruction system for the NASA/GSFC ground data transmission network, and of required ground facility modifications for the incorporation of data compression. The total time period of Phase II is 15 months.

During this reporting period, a new study area was added to the project. An analysis is to be performed to investigate differences in requirements for computer processing and real-time computer handling of compressed telemetry data and of standard uncompressed data.

1.1 PURPOSE AND SCOPE

The purpose of this report is to discuss the work performed during the second 3 months of Phase II and to present the program of activity for the next 3 months. The report has nine sections. Section 1 introduces the project and the report. Sections 2 through 5 present a detailed discussion, by task item, of the work performed during the reporting interval. Section 6 describes new concepts and techniques developed during the interval. The work planned for the next reporting interval is described in Section 7. Conclusions and recommendations are given in Section 8, and the bibliography is in Section 9.

1.2 PROJECT

Phase II, which includes an evaluation of techniques, a noise study, a computer utilization study, and a system conceptual design study, is divided into seven tasks, beginning with Task VII. The new tasks, XII and XIII, are scheduled to begin during the next reporting period.

1.2.1 Task VII, S-49 Compression Model Analysis

Task VII involves an extension of the compression technique evaluation by computer simulation under Phase I, using telemetry data from a more recently-launched satellite than the S-6. The data used are from the S-49, OGO-A satellite.

1.2.2 Task VIII, New Data Compression Model Evaluation

Task VIII evaluates, by computer simulation, new data compression techniques which were developed late in the first phase of the study. These techniques include sample selectors as well as promising methods of buffer feedback for the purpose of queuing control.

1.2.3 Task IX, Theoretical Buffer Analysis

Task IX involves a study of adaptive buffer fullness control on a purely theoretical basis. The theoretical evaluation of new feedback methods before computer simulation will materially increase the efficiency of the search for new techniques. The evaluation of existing methods will lend credence to the results of tests already performed.

1.2.4 Task X, Data Compression Mechanization Study

This task is concerned with the problems of incorporating a data compression and reconstruction system into the NASA/GSFC ground information transmission network. It is divided into three major subtasks.

1.2.4.1 System Configuration Analysis. This subtask investigates existing and projected NASA/GSFC data acquisition stations, control center stations, and data transmission requirements, to determine (1) the configuration of a data compression and reconstruction system for the ground data transmission network, and (2) the required ground facility modifications for the incorporation of this system.

1.2.4.2 Secondary Data Compression. This subtask studies the effects of compressing data which has been previously compressed, assuming that secondary compression is implemented by a higher-order compression algorithm.

1.2.4.3 Transmission Link Simulation. This subtask investigates, by computer simulation, the effect of ground transmission link noise on compressed data. The effect on the reconstructed data of noise added to the compressed data transmission link bit stream will be compared with the effect of the same noise added to the uncompressed data bit stream.

1.2.5 Task XI, Theoretical Transmission Error Analysis

Task XI analyzes, on a theoretical basis, the effect of errors which can occur in a compressed data bit stream. The kinds of errors to be investigated include those in data amplitude, sensor identification, and sample times, and errors resulting from the loss of frame synchronization. An analysis of the effect of transmission errors investigated in Task X on various data compression systems will also be done as a part of Task XI.

1.2.6 Task XII, Real Time Computer Analysis

This task investigates the capability of a digital computer to handle real time compressed data for spacecraft status checks and experiment monitoring. A comparison is to be made between the times required to handle telemetry data in a compressed format and those necessary to handle the same data in the present standard format.

1.2.7 Task XIII, Data Processing Computer Analysis

Task XIII compares computer operating times and other requirements necessary to process telemetry data in a compressed format with requirements for processing the same data in the present standard format.

1.3 BACKGROUND

During the first phase of this study, a preliminary investigation was made of the relative effectiveness of known data compression techniques as applied to GSFC satellite telemetry data.

The following steps were taken:

- Literature search
- Compression model selection
- Compression model and buffer simulation
- Time and channel identification study
- Proposal of optimum compression models

1.3.1 Literature Search

The initial step was a search of international literature on data compression. A list of over 30 different methods was compiled; each method was critically examined to determine its probable merits.

1.3.2 Compression Model Selection

The second step was an elimination process to reduce the number of models found during the literature search to a small group containing those showing the greatest promise of high performance. Six data compression models were placed in this group for subsequent testing. These included five polynomial sample selectors and one bit plane encoding technique.

1.3.3 Compression Model and Buffer Simulation

The next step, constituting the bulk of the work, was divided into two parts: (1) a compression model simulation analysis and (2) an adaptive buffer queuing control study.

1.3.3.1 Compression Model Simulation Analysis. Using an IBM 7094 digital computer, simulation analyses were made of the compression models chosen in the second step. S-6 data waveforms,used in this simulation, were selected to match particular data classes specified by GSFC. The results of these tests showed that a particular type of first-order polynomial interpolator achieved the highest performance in reducing the bandwidth of the S-6 telemetry data. This interpolator is capable of fitting straight lines to the sampled data waveform by adjusting their end points with four degrees of freedom.

Another part of the simulation analysis investigated the effects of precompressed filtering and adaptive techniques on compression efficiency and on reconstructed data fidelity. The results of this investigation indicated that both adaptive and nonadaptive precompression filtering may effectively enhance data compressor performance. This portion of the study, however, was too brief to be conclusive.

1.3.3.2 Adaptive Buffer Queuing Control Study. In this study, simulation tests were performed to determine the type of queuing control, if any, that would be needed to curtail or eliminate data loss from buffer overflow in a telemetry data compressor for an S-6 satellite. The most important result of this study was that, because of the data stationarity, adaptive queuing control would not be required for the S-6 data as long as the buffer readout rate exceeded the average readin rate by a factor of at least 1/0.98. However, to guard against the possibility of an abnormally low readout rate setting, a control system which monitored queue length was found to be necessary.

1.3.4 Time and Channel Identification Study

Relative efficiencies of several methods of time and channel coding were investigated. It was found that the transmission of a minor frame channel identification work with each data sample was the most efficient coding technique for S-6 data among those studied. With this method, at least one data/channel word in each minor frame must be transmitted in order to maintain a time reference and to identify subcommutated sensors.

1.3.5 Proposal of Optimum Compression Models

The final step consisted of a tradeoff analysis of the six compression models tested. The models which demonstrated the highest overall performance were tentatively proposed for compressing S-6 and similar data.

The spread in compression model performance, resulting from the tradeoff analysis, was surprisingly low. The first-order four-degree-of-freedom interpolator mentioned above was proposed for the groundbased data compression applications and, because this model is relatively complex, a simpler – though less effective – first-order interpolator was proposed for space applications. Queuing control, time and channel coding, and reconstruction methods were proposed for both applications.

As the study progressed, it became apparent that the effectiveness of existing compression techniques on S-6 data was not as high as had been experienced previously at LMSC on other data. A typical example of the overall bandwidth compression ratios obtained on the S-6 data is 3.3:1. A number of methods for improving the compression were proposed, including partial data processing before transmission (signal reduction) and methods to take advantage of the periodicity which exists in many of the S-6 data waveforms. It was estimated that with the proper application of these techniques, bandwidth compression ratios exceeding 10:1 can be achieved on S-6 data. Therefore, recommendations for future study included investigations of these data compression techniques.

1.4 SUMMARY OF WORK PERFORMED

1.4.1 Task VII, S-49 Compression Model Analysis

Activity on this task during the second quarter of Phase II consisted of an exploratory multisensor compression and buffer simulation run on 184,000 input data samples. A sample compression ratio of 13.4 was obtained from the run, using the first-order disjoined interpolator. The run provided an excellent indication of buffer queue behavior in the absence of control; with the ratio of average input rate to readout rate equal to 0.98, the maximum queue length was 505 samples.

1.4.2 Task VIII, New Data Compression Model Evaluation

Computer programs were written and debugged for four new or modified sample selection techniques during the reporting period. They are: exponential interpolation, short-term frequency component selection, bit plane encoding, and broadside data compression. In addition, a new queuing control test model designed to evaluate adaptive filtering and sample deletion with Goddard Space Flight Center telemetry data was programmed and compiled. The model operates within the framework of the simulated multisensor compression and reconstruction systems which were programmed during Phase I.

1.4.3 Task IX, Theoretical Buffer Analysis

One of the results of the theoretical study of buffer queuing control (Ref. 1) was that a combination of the queue length and its time integral promises to be a particularly effective parameter to control the data tolerance. During the second quarter, a control system using this parameter was programmed as the first step in its computer evaluation.

1.4.4 Task X, Data Compression Mechanization Study

During the second quarter, as in the first, all activity was concentrated on the transmission link simulation subtask of Task X. All simulation programs were completed, and production test runs were begun. The method and range of Pareto noise parameter variation was also established.

Section 2
TASK VII, S-49 COMPRESSION MODEL ANALYSIS

During the reporting period, an exploratory multisensor compression and buffer simulation computer run, without queuing control, was made on an 84-second segment of the S-49 data, using the FOIDIS selector. Approximately 584,000 data samples were processed. This relatively long run was made primarily to locate a short segment (about five seconds) of S-49 multiplexed data which exhibited a changing activity, so that meaningful queuing control runs can be made. The run also produced a C-Tape that contains enough data samples to provide a reasonable statistical sample for the transmission link noise study.

The tolerances used and the compression ratios obtained are shown for each sensor in Table 2-1. The combined sample compression ratio (the number of samples processed divided by the number of samples retained by the selector) obtained for the run was 13.4. A tolerance of 0.01 data units was assigned to all digital sensors, which guarantees full accuracy on these sensors. Note that "sensors" 70 through 78 are in reality sync, time, and status words. These channels were also assigned tolerance values of 0.01 data units, to make sure all binary words could be reproduced after reception with 9-bit accuracy. All sensors which were programmed off during the run were assigned a tolerance of 1,000, which exceeds the full-scale value of 255 data units for the analog sensors and 511 for the digital sensors. All other analog sensors were assigned what were considered to be reasonable values of tolerance. The sensor names which correspond to their numbers shown in Table 2-1 in this report are given in Table 2-2 of Ref. 1.

With the buffer sample readout rate set so that the ratio of average input rate to readout rate was 0.98, the queue reached a maximum of 505 samples in the infinite-capacity simulated buffer at one point in the run. During the 84-second period the buffer completely emptied, following a queue buildup, a total of five times. Each buildup was to at least 225 samples. The variable data activity which caused this queue length fluctuation exhibited, in part, a cyclic pattern which appeared to be synchronized with the satellite roll.

Table 2-1

S-49 MULTISENSOR COMPRESSOR SIMULATION RUN RESULTS

Run designation:	S-49-03
Selector:	FOIDIS
Combined sample compression ratio:	13.4
Data tape:	GSFC 1125
Time period:	266−350 sec from start of tape
Maximum queue length:	505; $\bar{\rho}$ = 0.98*
No queuing control	

Experiment Number	Sensor Number	Minor Frame Channel No.	Analog or Digital	Tolerance (data units or counts)**	Sample Compression Ratio
1	1	81, 82, 83, 84	D	0.01	13.7
2	2	4	A	2	99.3
	3	5	A	2	∞
	4	6	A	2	222.1
	5	7	A	2	518.4
	6	21	A	2	76.5
	7	22	A	2	24.7
	8	23	A	4	99.3
3	9	10, 42, 74, 106	A	1000	∞
	10	36, 107	A	1000	∞

*$\bar{\rho}$ = average buffer input sample rate/output rate.

**Full scale data units = 511 for digital sensors, and 255 for analog sensors.

Table 2-1 (Cont'd)

Experiment Number	Sensor Number	Minor Frame Channel No.	Analog or Digital	Tolerance (data units or counts)**	Sample Compression Ratio
4	11	55	A	4	311.1
	12	56	A	4	∞
	13	57	D	0.01	∞
5	14	13, 77	A	2	9.9
	15	25, 89	D	0.01	4.5
	16	104	A	8	359.0
6	17	58	D	1000	∞
	18	59	D	1000	∞
	19	60	D	1000	∞
7	20	19	D	0.01	1.0
	21	20	A	2	358.9
	22	28, 52	D	0.01	1.0
	23	29	A	2	141.4
	24	49	D	0.01	1.1
	25	50	D	0.01	27.6
	26	51	D	0.01	1.6
	27	53	A	2	22.8
8	28	114	D	0.01	4.0
	29	115	D	0.01	1.0
9	30	43, 90, 113	D	1000	∞
	31	100	A	1000	∞
	32	116, 121, 122, 123, 124	D	1000	∞
10	33	14, 37, 61, 86, 117	A	2	281.1
	34	15, 38, 62, 87, 118	A	2	77.0

Table 2-1 (Cont'd)

Experiment Number	Sensor Number	Minor Frame Channel No.	Analog or Digital	Tolerance (data units or counts)**	Sample Compression Ratio
	35	16, 39, 63, 88, 119	A	2	143.1
	36	80	A	2	99.3
11	37	30, 94	A	2	39.0
	38	31, 95	A	2	78.4
	39	32, 96	A	2	∞
	40	91	A	2	41.3
	40	92	A	2	36.2
	42	93	A	2	∞
	43	102	A	2	∞
12	44	125	A	1000	∞
	45	126	A	1000	∞
	46	127	A	1000	∞
	47	128	A	1000	∞
13	48	9, 12, 41, 44, 73, 76, 105, 108	D	0.01	6.2
15	49	24	A	4	∞
	50	27	A	4	11.4
	51	40	A	4	141.4
	52	54	A	4	14.8
	53	68	A	4	22.5
	54	69	A	4	∞
	55	70	A	4	26.4
	56	71	A	4	666.6
	57	72	A	4	63.9
	58	109	A	4	25.8
17	59	46, 110	A	1000	∞
	60	47, 111	A	1000	∞

Table 2-1 (Cont'd)

Experiment Number	Sensor Number	Minor Frame Channel No.	Analog or Digital	Tolerance (data units or counts)**	Sample Compression Ratio
	61	48, 112	A	1000	∞
18	62	8, 26, 45, 64 85, 103, 120	A	1000	∞
19	63	78	A	2	6.3
	64	79	A	2	13.1
20	65	11	D	0.01	933.2
	66	17	D	0.01	∞
	67	18	D	0.01	933.2
OPEP 2	68	75	A	4	933.2
SPARE	69	101	—	1000	∞
SYNC	70	1	D	0.01	∞
	71	2	D	0.01	∞
	72	3	D	0.01	∞
TIME	73	33	D	0.01	518.4
	74	34	D	0.01	1555.3
	75	35	D	0.01	27.6
STATUS	76	65	D	0.01	3.3
	77	66	D	0.01	933.2
	78	67	D	0.01	∞

Section 3
TASK VIII, NEW DATA COMPRESSION MODEL EVALUATION

The evaluation of new compression models divides logically into two categories – new selector evaluation and the investigation of new queuing control techniques. The new selectors are being tested on a single-sensor basis, while the queuing control evaluation is being done with telemetry data in their original multiplexed formats. Configurations of the test models were established during this reporting period; they are described in the following paragraphs.

3.1 NEW SELECTOR EVALUATION

Computer programs have been written and debugged for four new or modified compression models. They are: (1) exponential interpolator, (2) short-term frequency component selector, (3) bit plane encodor, and (4) broadside data compressor.

3.1.1 Exponential Interpolation

3.1.1.1 Principle. The principle of exponential interpolation is similar to that of polynomial interpolation, differing only in the use of exponential reference patterns instead of polynomial reference patterns. The process of exponential interpolation is diagrammed in Fig. 3-1, where its operation is shown in a hypothetical situation. As can be discerned from Fig. 3-1, the exponential interpolation process consists of fitting the longest possible exponential waveform to the data, such that a given peak error criterion between the data and this exponential waveform is not exceeded.

Although this process is very similar to that of polynomial interpolation, at least one significant complication must be considered. Not just one, but two, different classes of exponential reference patterns exist, and should be considered simultaneously.

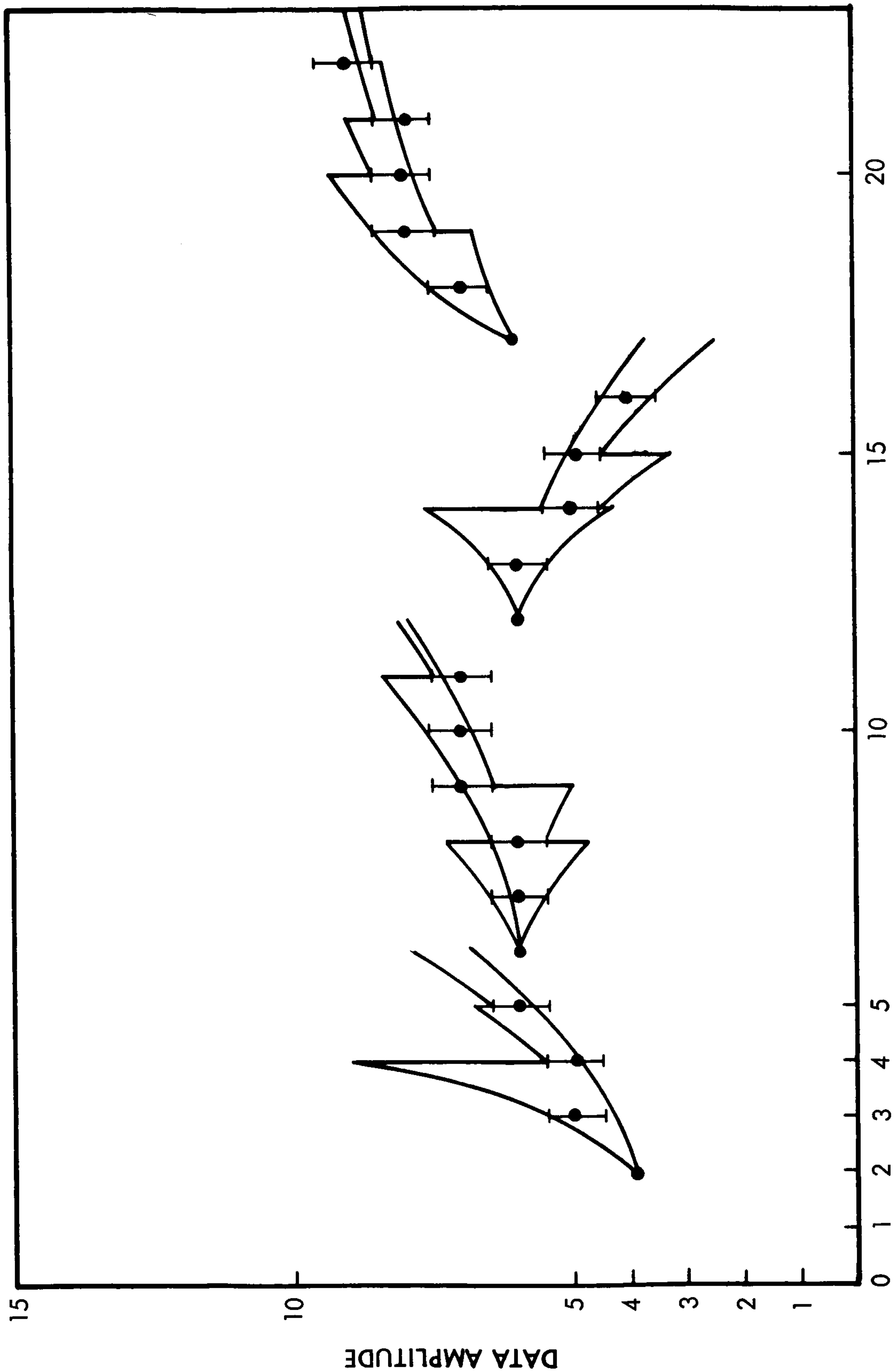


Fig. 3-1 Operation of the Exponential Interpolator

These two classes of reference patterns are shown in Fig. 3-2. Clearly the first class of reference patterns consists of growing exponential waveforms, while the second class of reference patterns consists of asymptotic exponential waveforms. A general expression which describes the members of the first class of reference patterns is:

$$y_n = a_1 \pm b_1 u_1^n ; (u_1 > 1), n = 0, 1, 2, \ldots \qquad (3.1a)$$

A general expression which describes the members of the second class of reference patterns is:

$$y_n = a_2 \pm b_2 (1 - u_2^n) ; (u_2 < 1), n = 0, 1, 2, \ldots \qquad (3.1b)$$

where

y_n = n^{th} sample magnitude

a_1, b_1, u_1, a_2, b_2, and u_2 = constants which establish the exact pattern shape

In each case, the sign of the second term determines whether the waveform increases or decreases with increasing n.

In order to specify a particular member of either class of exponential waveforms, three quantities must in turn be specified; namely, a, b, and u. If it is desired to fit an exponential waveform from a given class through two given data values, there exist, in general, an infinite number of possible combinations of a, b, and u which achieve the desired fit. However, if one of these three quantities, such as b, is specified a priori, the values of the remaining two quantities a and u which achieve the desired fit are specified uniquely.

3.1.1.2 Simulated Exponential Interpolator. A prespecified value for b is utilized in the exponential interpolator which has been simulated by means of digital computer, allowing unique calculations of a and u in all cases.

The macroscopic operation of an exponential interpolator, utilizing both classes of reference patterns simultaneously, is described by the flow chart of Fig. 3-3. Essentially, this exponential interpolator consists of two serial, independent interpolators. The first interpolator utilizes members of the class of growing exponential waveforms

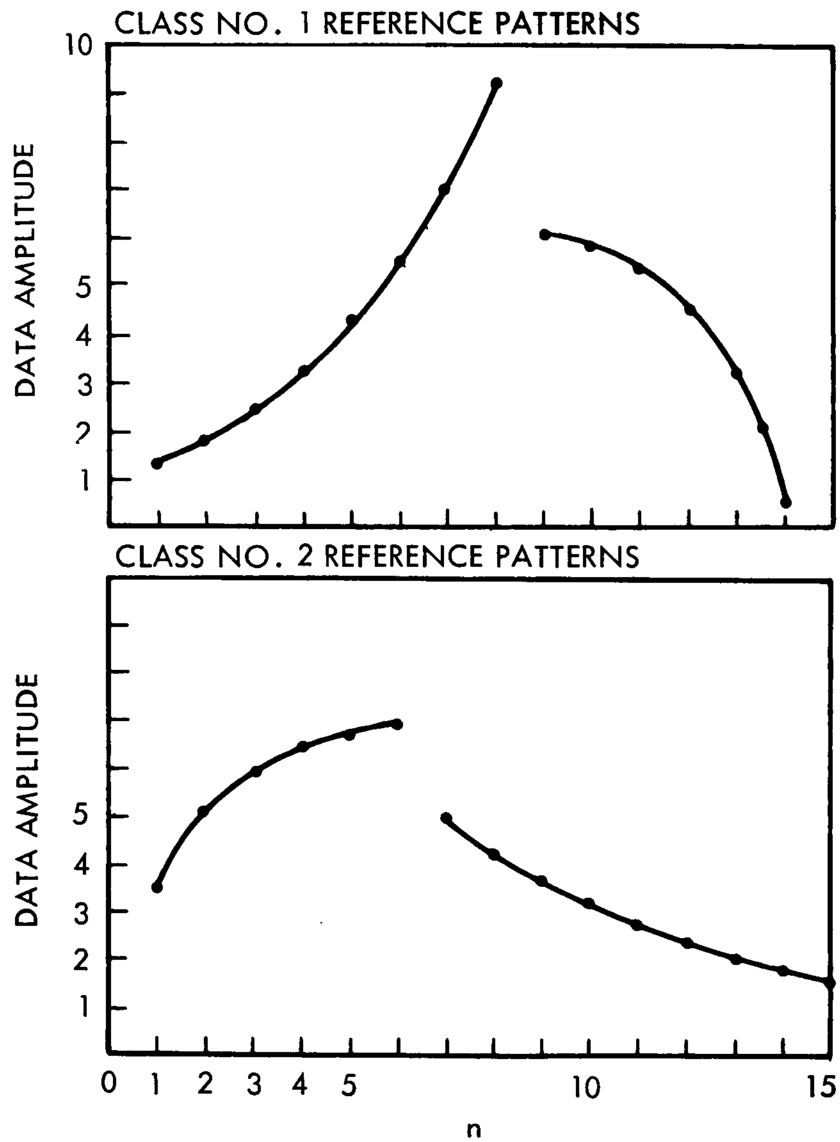


Fig. 3-2 Exponential Reference Patterns

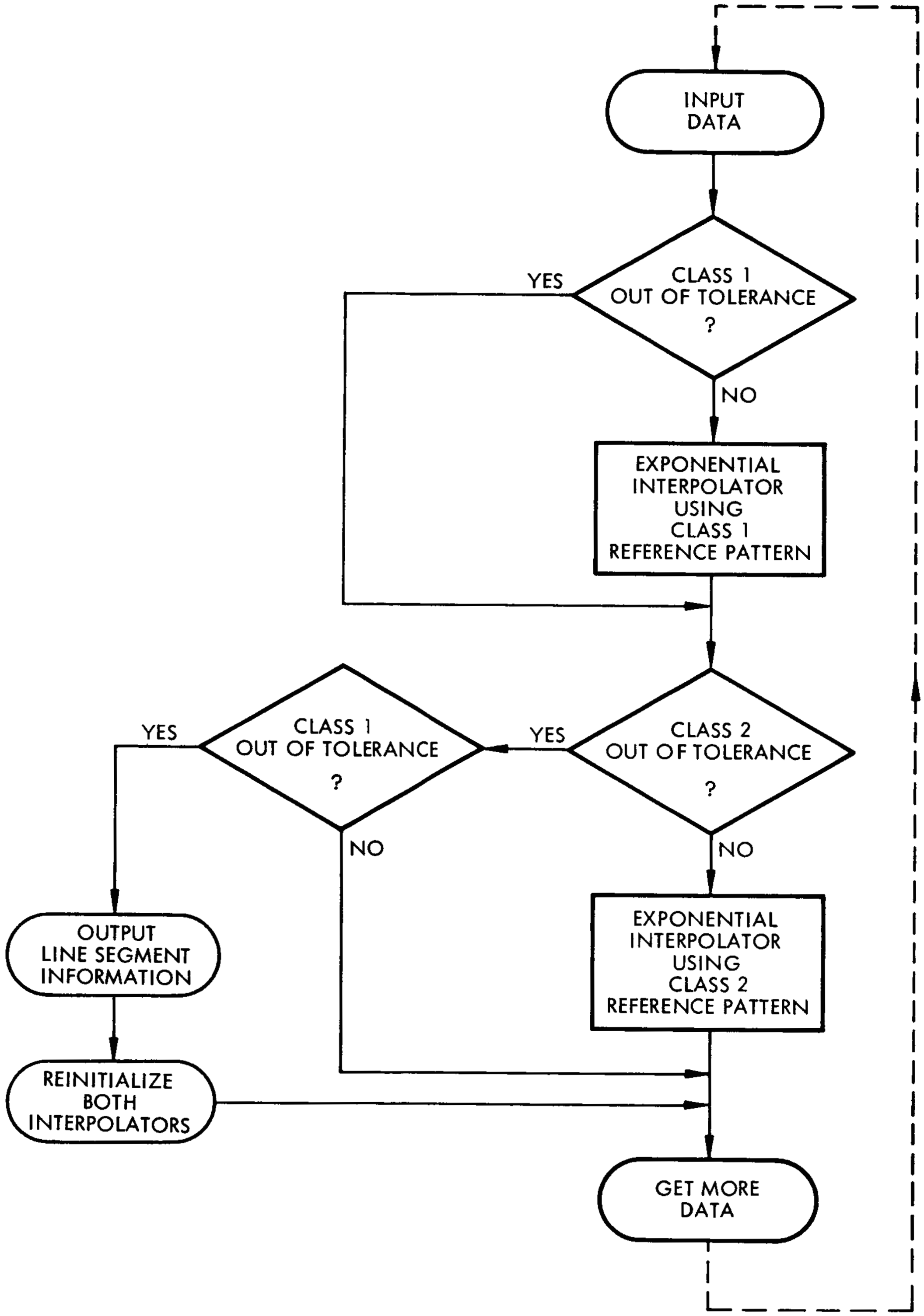


Fig. 3-3 Flow Chart of Exponential Interpolator

as reference patterns, while the second interpolator utilizes members of the class of asymptotic exponential waveforms as reference patterns. Both interpolators independently process all data points until one or the other of the interpolators encounters an out-of-tolerance condition. This interpolator temporarily discontinues operation while the remaining interpolator continues to process incoming data. When this remaining interpolator finally encounters an out-of-tolerance condition, information is "transmitted," describing a reconstructable line segment. This information consists of the time of occurrence of the final out-of-tolerance condition, the initial line segment amplitude, the averaged value of the previous upper and lower limits of the last interpolator to encounter an out-of-tolerance condition, as well as information identifying the last interpolator to encounter an out-of-tolerance condition. An exponential waveform can be reconstructed on the basis of this information, such that a peak error criterion between the original data and the reconstructed waveform is not exceeded.

3.1.1.3 Performance. Depending upon the a priori choice of the value of parameter b, the exponential waveforms used as reference patterns take on a more or less exponential shape. Large values of b cause both classes of reference patterns to contain approximately linear waveforms over the range of line segment length normally encountered. Similarly, small values of b cause both classes of reference patterns to contain highly exponential waveforms over the range of line segment length normally encountered. Appropriate prespecification of b, therefore, causes the behavior of the exponential interpolator to range from that of first order interpolation in the limiting case to that of marked exponential interpolation. An interesting facet of the computer-evaluation runs planned for the exponential interpolation will be a comparison of this disjoined interpolator with the FOIDIS selector, when b is very large.

3.1.2 Short-Term Frequency Component Selection

3.1.2.1 Principle. With this selector, a Fourier analysis is performed on a prescribed number of most recent data samples, the analysis being updated as new data samples are received. This short-term frequency analysis is then used to decide whether or not the most recent data sample is to be retained. If a significant fraction of the total short-term data energy is present at relatively high frequencies, the most recent data sample is likely to be retained. If a significant fraction of the total short-term data energy is present at relatively low frequencies, the most recent data sample is likely to be discarded. This sample selection technique tends to maintain the short-term sampling rate of the data in proportion with the short-term information bandwidth of the data. This selector provides no peak error guarantee. Its operation is further described in Appendix II.

3.1.2.2 Simulated Short-Term Frequency Component Selector. A computer program, based on Eq. (II. 1) and (II. 2) of Appendix II, has been written to simulate the operation of this selector on a single sensor of Goddard satellite data. This simulated selector operates by comparing the normalized cumulative power density $C_N(M)$ ($M = 0, 1, 2 .. N$) with a prescribed threshold, when $C_N(M)$ is calculated according to Eq. (II. 2). Let M_T be defined as the smallest value of M for which $C_N(M)$ is greater than this prescribed threshold. Also, let N_p be defined as the number of samples which have been processed since the last retained data sample (including the processing of the present data sample). The decision rule used in retaining and deleting data samples is as follows:

$$\text{if } \left[{}^{N}/{}_{M_T}\right] \begin{pmatrix} > \\ \leq \end{pmatrix} N_p, \text{ then } \begin{pmatrix} \text{delete} \\ \text{retain} \end{pmatrix} \text{ the present data sample.}$$

($\left[\quad\right]$ denotes the "integer part of"). Specific examples of the use of this decision rule are shown in Fig. 3-4.

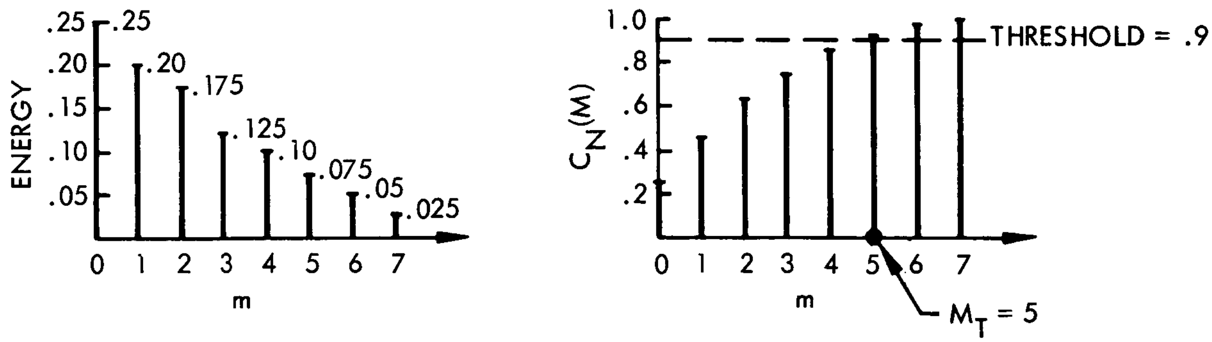
Reconstruction of such compressed data is accomplished by interpolating between retained data samples either linearly or according to a zero-order hold (stair-step) method. The error statistics compiled between the original and reconstructed data include an error frequency distribution as well as mean and rms error.

3.1.3 Bit Plane Encoding

The bit plane encoding technique employed during Phase II of this contract differs in many important respects from that technique employed during Phase I. Correspondingly, the computer program used to simulate the bit plane encoding of single-channel telemetry data, which was written during Phase I, was rewritten to implement and take

N = 7 ; 2N + 1 = 15 DATA SAMPLES INCLUDED IN FOURIER ANALYSIS

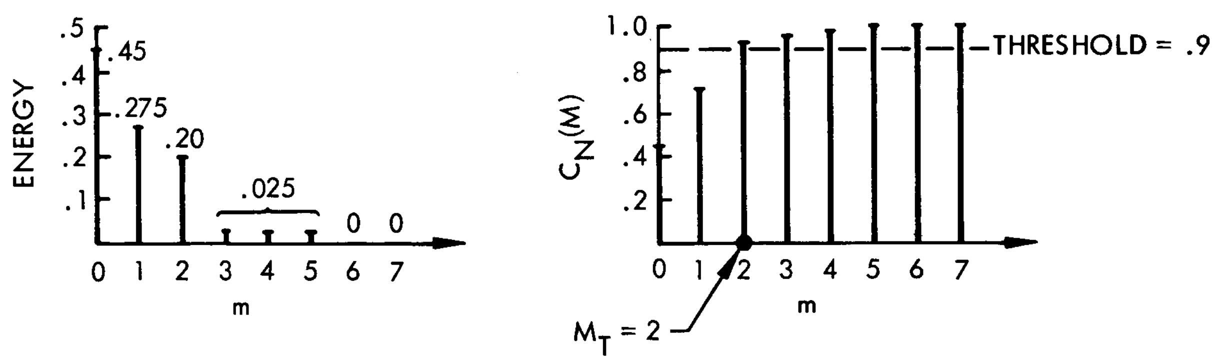
EX. 1) NORMALIZED ENERGY SPECTRUM — NORMALIZED CUMULATIVE ENERGY SPECTRUM



$[N/M_T] = [7/5] = [1.4] = 1$

∴ THE PRESENT DATA SAMPLE IS <u>RETAINED</u>

EX. 2) NORMALIZED ENERGY SPECTRUM — NORMALIZED CUMULATIVE ENERGY SPECTRUM



$[N/M_T] = [7/2] = [3.5] = 3$

∴ THE PRESENT DATA SAMPLE IS DISCARDED IF AND ONLY IF 2 OR FEWER SAMPLES HAVE BEEN PREVIOUSLY PROCESSED SINCE THE LAST RETAINED DATA SAMPLE.

Fig. 3-4 Short-Term Frequency Component Selector Operation

advantage of the experience gained in the performance of that portion of Phase I dealing with bit plane encoding. The significant differences between the original and revised computer programs and associated encoding techniques are summarized below. (See Appendix III of Ref. 2 for applicable terms and definitions.)

- Since the performance of any bit plane encoding technique is strongly dependent upon the size of blocks which are encoded, block size is a parameter which should be investigated. The revised computer program now allows encoding of data in blocks of size $2^n; n = 1, 2, \ldots,$ whereas the original computer program only allowed encoding of data in blocks of size 128 ($= 2^7$).
- When using a cyclic Gray code rather than a simple binary code for encoding data words, the revised computer program encodes only that portion of the data words remaining after deletion of the bits included in the Class C planes. The original computer program encoded entire data words prior to deletion of the bits included in the Class C planes.
- The code words, code word groups, and special words used by the revised program in encoding are much more efficient than those shown in Table III-1 of Ref. 2. These revised code words and code word groups are described in Table 3-1. No attempt has been made to predicate these code word groups on any particular transmitted word length. Although additional dummy bits would occasionally be needed to allow the transmission of complete words in the case of any given transmitted word length, these bits would not significantly affect the overall compression ratio.

Bit-by-bit transmission is intiiated when the count of changes for a given Class B plane exceeds $[M_B/N_B]$, where [ ] is used to denote the "integer part of." This decision is motivated by the requirement that the number of bits necessary to describe the location of transitions in a given Class B plane (i.e., number of transitions $\times N_B$) not exceed the number of bits in the entire Class B plane (i.e., M_B).

A typical sequence of code words and groups might be as follows (the letters refer to the case description used in Table 3-1):

abcde abcd abcd abcde abcde

Table 3-1

CODE WORDS AND CODE WORD GROUPS

Case	Words and Groups	Definition
(a)	$a_1\ a_2\ a_3 \cdots a_9$	$(a_i = 0, 1)$; $a_i = 1$ indicates that the i^{th} bit plane is an A plane.
(b)	$b_1\ b_2\ b_3 \cdots b_9$	$(b_i = 0, 1)$; b_i describes the first bit in the i^{th} bit plane.
(c)	$c_1\ c_2\ c_3 \cdots c_9$	$(c_i = 0, 1)$; $c_i = 1$ indicates that the i^{th} bit plane is to be transmitted bit by bit.
(d)	$(N_A\ N_B)$ bits	Those bits necessary to describe the locations of the transitions in B planes, where N_A = total number of transitions in the given block, and $N_B = \log_2$ (block size).
(e)	$(M_A\ M_B)$ bits	Those bits contained in the B planes which are transmitted bit by bit, where M_A = total number of bit planes transmitted bit by bit, and M_B = block size.

This sequence may be interpreted as follows:

- First block – some Class B planes transmitted bit by bit
- Second and third block – no Class B planes transmitted bit by bit
- Fourth and fifth block – some Class B planes transmitted bit by bit

Such a sequence is not entirely unambiguous, but the addition of a minimal number of "comma words" would easily make it so. These "comma words" would not significantly affect the overall number of bits required, however. Consequently, their contribution is not included when calculating compression ratios achieved by encoding via the revised program.

3.1.4 Broadside Data Compression

Because of the modulation due to satellite spin, many Goddard satellite data waveforms are highly periodic. This periodicity is added to normal slow data variations caused by the orbital movement of the vehicle. Such data are very difficult to compress, although they are inherently highly redundant. One way to cope with periodic waveforms, such as those caused by spin of the satellite vehicle, is to separate data samples which occur at corresponding portions of succeeding waveform cycles into groups, and then to compress, concurrently and independently, each of the groups. Reconstruction can be accomplished by first restoring the missing samples in each group, and then reassembling the waveform by interlacing the groups. In this way, the cyclic nature of the waveform, as well as its phase, would be preserved.

Concurrent action on all portions of the waveform cycle is a "broadside" approach to the problem of redundancy removal from cyclic waveforms. The technique is thus called broadside data compression. Its implementation could be accomplished by synchronizing the multiplexer to the roll period of the satellite. If synchronous multiplexer operation is difficult or impossible, it is feasible to operate the multiplexer asynchronously, but to group together the corresponding data samples by means of a post-multiplexer sorting process. This can be done with reference to appropriate aspect information.

A computer program has been written to process data from a number of sensors (including the aspect sensor) and to sort these data samples into groups, each containing data samples from corresponding portions of consecutive waveform periods. Using this program, the data samples from each of these groups are written onto a separate file on magnetic tape in standard T-3 format. This tape can then be processed as if each group was data from an individual sensor (pseudosensor).

In the case of the Goddard S-6 satellite, data from the aspect sensor exhibit a marked peak once each roll period. When using this program, data from this aspect sensor can be thresholded, and arrival of the peak can be used to define the onset of each roll period.

3.2 NEW QUEUING CONTROL EVALUATION

During the reporting period, a new queuing control test model program, designed to evaluate adaptive filtering and sample deletion with GSFC telemetry data, was written and compiled. The model operates within the framework of the simulated multisensor compression and reconstruction systems which were programmed during Phase I. The simulated system is capable of employing digital exponential filtering, digital running-average filtering, and sample deletion, each on an adaptive, individual sensor basis. A block diagram of the system is shown in Fig. 3-5.

3.2.1 Exponential Filtering

The general expression for the digital exponential filter on a given sensor's sampled waveform is

$$\hat{y}_j = ay_j + b\hat{y}_{j-1}$$

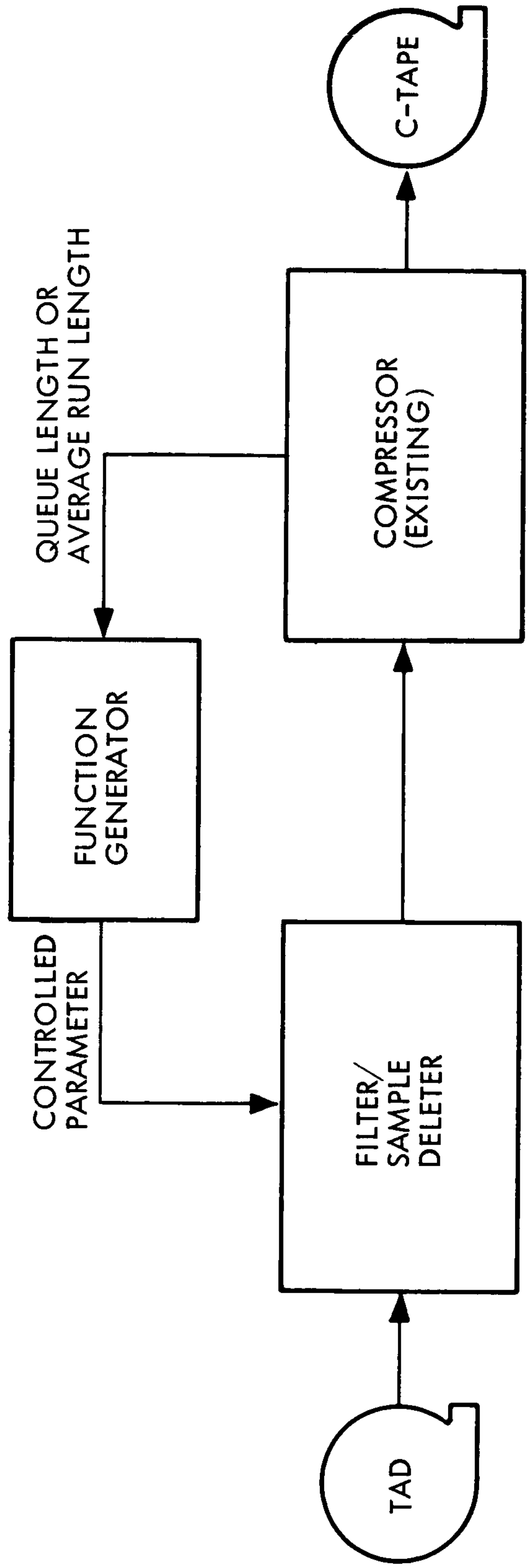


Fig. 3-5 Adaptive Filtering System Block Diagram

where

$\hat{y}_j$ = filtered j^{th} data sample

y_j = unfiltered j^{th} data sample

$\hat{y}_{j-1}$ = filtered $j - 1^{st}$ data sample

b = $1 - a$ = exponential filtering factor

Note that

$$a + b = 1$$

In a multisensor system, this filtering must be done on an individual sensor basis. The variable controlled by the monitored parameter is the filtering factor, b_i , $1 \le i \le S_T$, where S_T = total number of sensors. The factor b_i is made to vary in such a manner that the equivalent low-pass filter cutoff (half-power) frequency varies linearly with the monitored parameter.

As in the case of the queuing control systems programmed during Phase I (Ref. 2), the system monitors one of two parameters. The first parameter is the "short-term" compression ratio, obtained for a particular sensor by measuring the average run length of a given fixed number of most recent line segments. The second parameter is the buffer queue length. These variables are recalculated after each data sample.

3.2.1.1 Compression Ratio Monitoring System. The control curve for the compression ratio monitoring system is shown in Fig. 3-6. The control equations are

$$b_i = b_i' = e^{-\pi} e^{-2\pi \beta_{5i} (N_i - N_{Ti})} \qquad \text{if } e^{-\pi} \le b_i' \le b_{Mi}$$

$$b_i = 0 \qquad \text{if } b_i' < e^{-\pi}$$

$$b_i = b_{Mi} \qquad \text{if } b_i' > b_{Mi}$$

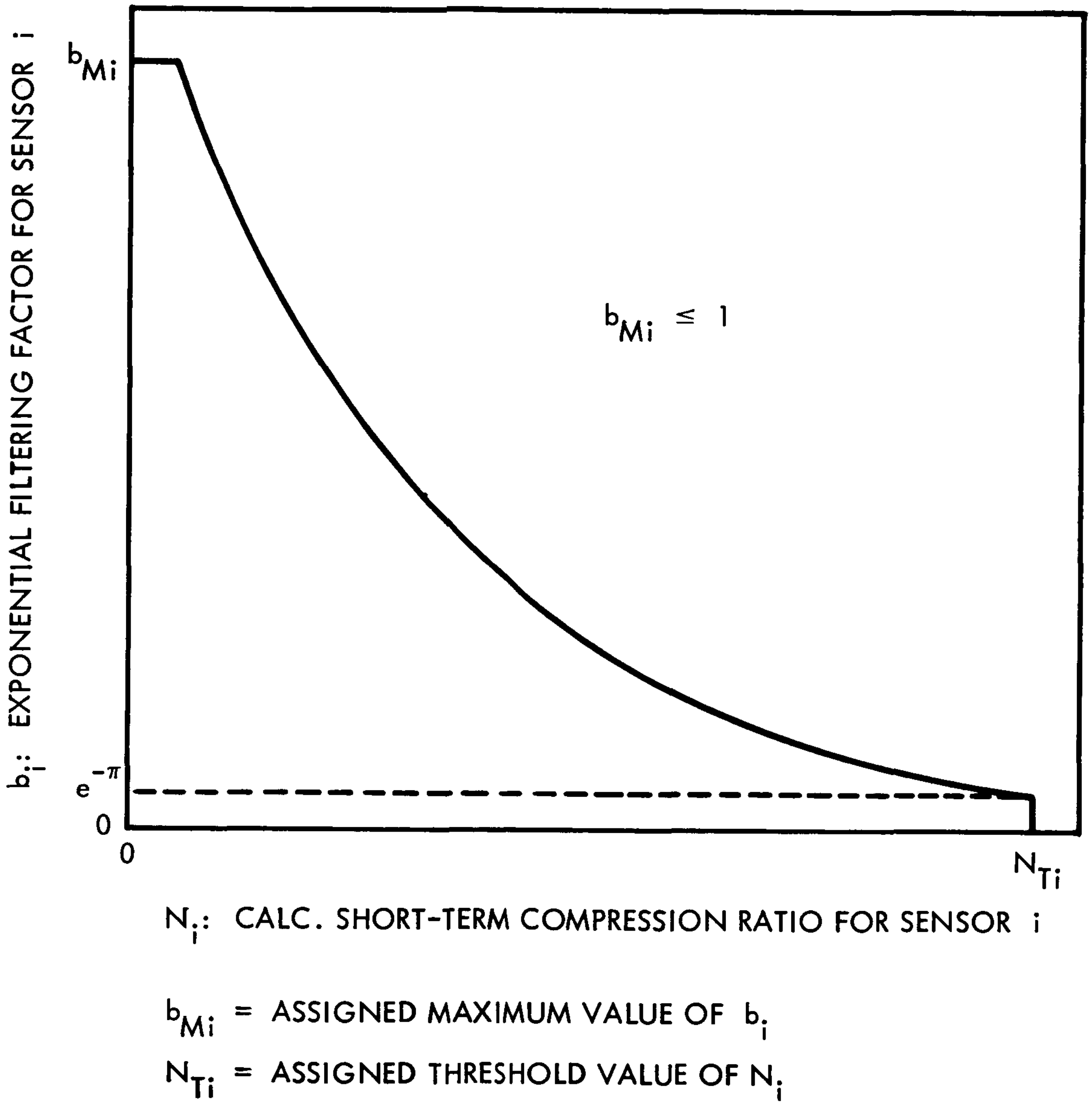


Fig. 3-6 Exponential Filtering Control Function, Compression Ratio Monitoring System

where

b_i = exponential filtering factor for Sensor i

β_{5i} = assigned feedback constant for Sensor i

N_i = calculated short-term compression ratio for Sensor i

N_{Ti} = assigned threshold value of N_i

b_{Mi} = assigned maximum value of b_i

The variables controlled by data cards are N_{Ti} , b_{Mi} , and β_{5i} , $1 \leq i \leq S_T$, where S_T = total number of sensors.

3.2.1.2 Queue Length Monitoring System. The control curve for the queue length monitoring system is shown in Fig. 3-7. The control equations are

$$b_i = b_i' = e^{-\pi} e^{2\pi \beta_{6i} [n - (n_{min})_i]} \qquad \text{if } e^{-\pi} \leq b_i' \leq 1$$

$$b_i = 0 \qquad \text{if } b_i' < e^{-\pi}$$

$$b_i = 1 \qquad \text{if } b_i' > 1$$

where

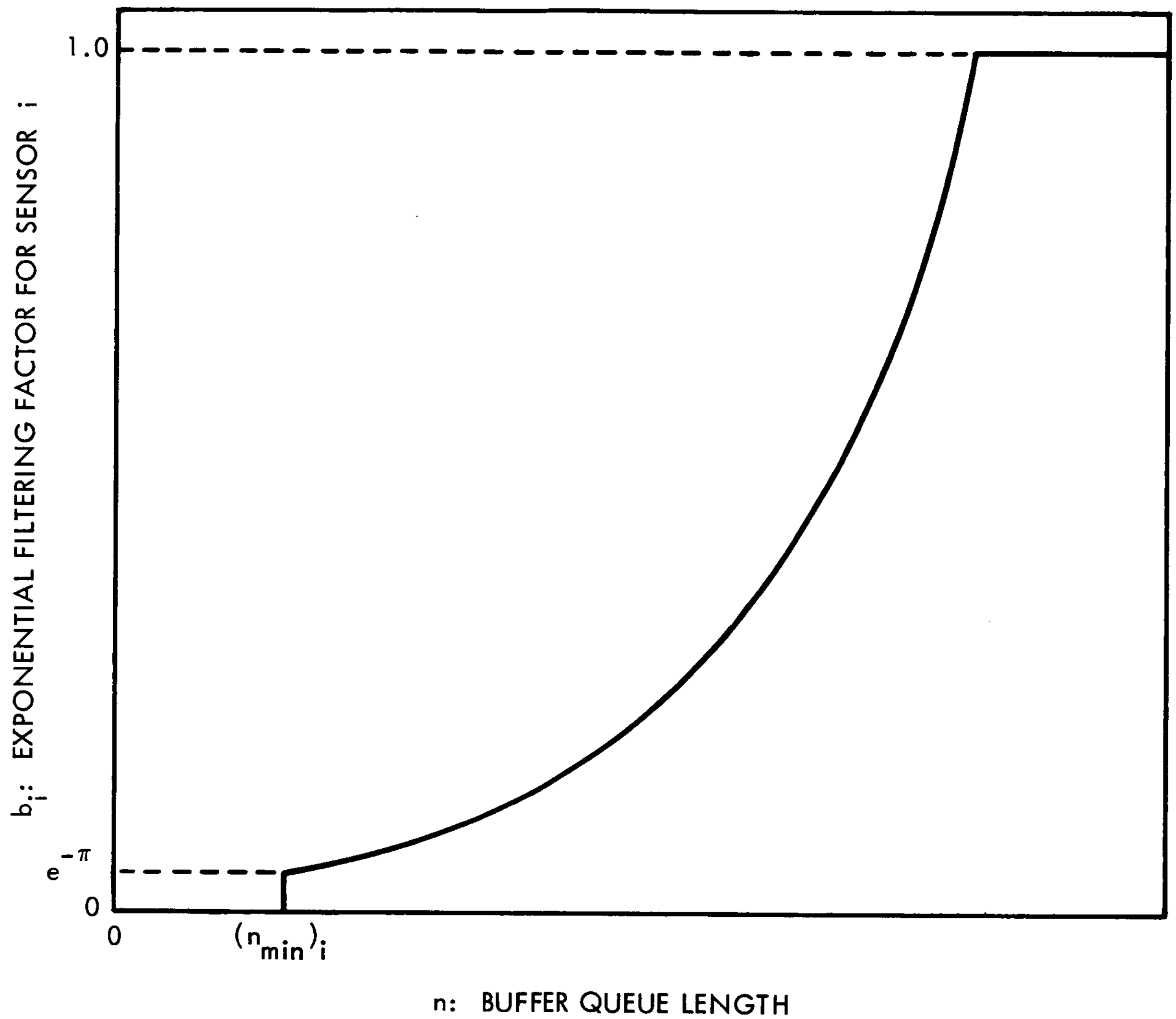
b_i = exponential filtering factor for Sensor i

β_{6i} = assigned feedback constant for Sensor i

n = buffer queue length

$(n_{min})_i$ = assigned threshold value of queue length for Sensor i

The variables controlled by data cards are β_{6i} and $(n_{min})_i$, $1 \leq i \leq S_T$, where S_T = total number of sensors.



$(n_{min})_i$ = ASSIGNED THRESHOLD VALUE OF QUEUE LENGTH FOR SENSOR i

Fig. 3-7 Exponential Filtering Control Function, Queue Length Monitoring System

3.2.1.3 Simulation Output. In addition to the output data available from the programs written during Phase I (Ref. 2, Par. IV.6), the exponential filtering program puts out the following data:

- Time histories of b_i, with start and stop time controllable, for any sensor (4020 plot)
- Histogram of b_i for any sensor (4020 plot)
- Average value of b_i for any sensor

3.2.2 Running-Average Filtering

The general expression for the running-average filter on a given sensor's sampled waveform is

$$\hat{y}_j = \frac{1}{T} \sum_{k=0}^{T-1} y_{j-k}$$

where

$\hat{y}_j$ = filtered j^{th} data sample

y_{j-k} = unfiltered $j - k^{th}$ data sample

T = number of samples averaged together

Filtering is done on an individual sensor basis. The variable controlled by the monitored parameter is T_i , $1 \leq i \leq S_T$. The quantity T_i is made to vary in such a manner that the equivalent low-pass filter cutoff frequency varies stepwise linearly with the monitored parameter. The program is written so that T_i never exceeds 20. As with exponential filtering, the system monitors either short-term compression ratio or queue length.

3.2.2.1 Sample Deletion Option. The running-average filtering system is capable of operating in two different modes:

- Filtering only – filter output sample rate equal to filter input sample rate
- Post-filtering sample deletion before compression – compressor input sample rate reduced by a factor of $1/T_i$ below the output rate of the filter; i.e., every T_i^{th} sample would be retained

3.2.2.2 Compression Ratio Monitoring System. The control curve for the compression ratio monitoring system is shown in Fig. 3-8. The control equations are

$$\frac{1}{T_i} = \frac{1}{T_i'} = 1 + \beta_{3i}(N_i - N_{Ti}) \qquad \text{if } \frac{1}{T_{Mi}} \le \frac{1}{T_i'} \le 1 \tag{3.2}$$

Calculate T_i from Eq. (3.2), and round off to the nearest integer.

$$T_i = 1 \qquad \text{if } \frac{1}{T_i'} > 1$$

$$T_i = T_{Mi} \qquad \text{if } \frac{1}{T_i'} < \frac{1}{T_{Mi}}$$

where

T_i = number of samples averaged together for Sensor i

β_{3i} = assigned feedback constant for Sensor i

N_i = calculated short-term compression ratio for Sensor i

N_{Ti} = assigned threshold value of N_i

T_{Mi} = assigned maximum value of T_i

The variables controlled by data cards are N_{Ti} , T_{Mi} , β_{3i} , $1 \le i \le S_T$.

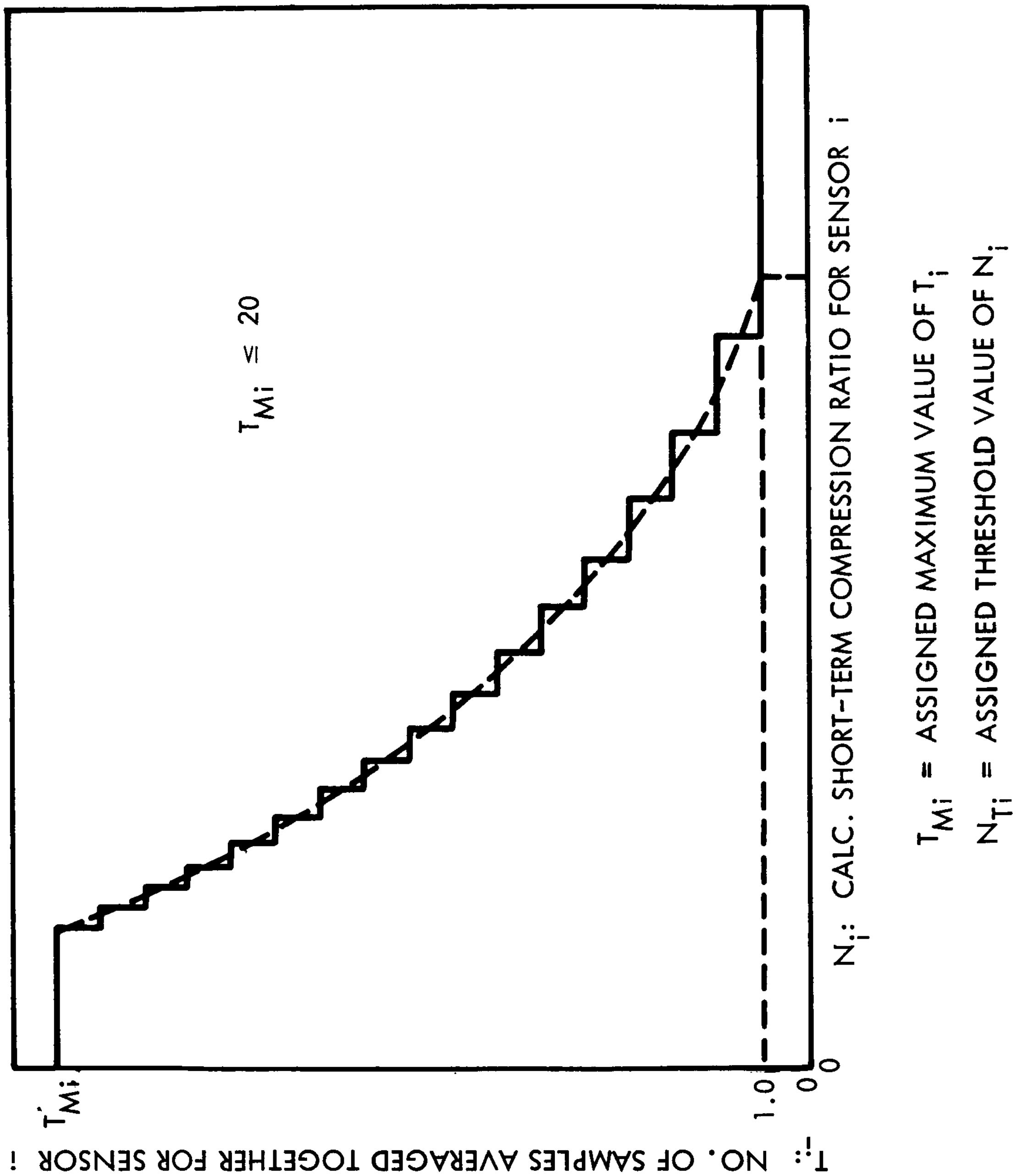


Fig. 3-8 Running-Average Filtering Control Function, Compression Ratio Monitoring System

3.2.2.3 Queue Length Monitoring System. The control curve for the queue length monitoring system is shown in Fig. 3-9. The control equations are

$$\frac{1}{T_i} = \frac{1}{T_i'} = 1 - \beta_{4i}\,[n - (n_{min})_i] \qquad \text{if } 0.05 \le \frac{1}{T_i'} \le 1 \tag{3.3}$$

Calculate T_i from Eq. (3.3), and round off to the nearest integer.

$$T_i = 1 \qquad \text{if } \frac{1}{T_i'} > 1$$

$$T_i = 20 \qquad \text{if } \frac{1}{T_i'} < 0.05$$

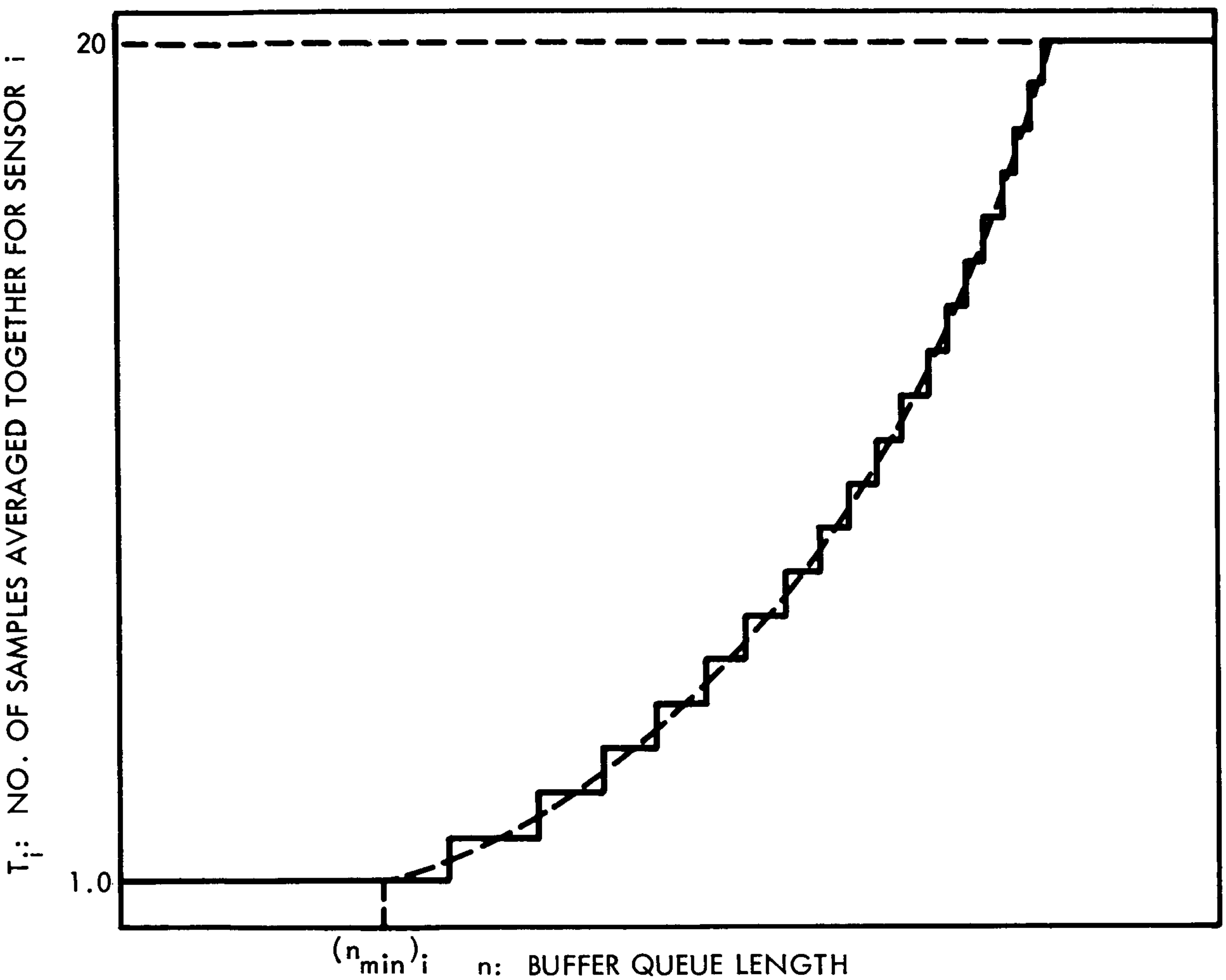
where

T_i	=	number of samples averaged together for Sensor i
β_{4i}	=	assigned feedback constant for Sensor i
n	=	buffer queue length
$(n_{min})_i$	=	assigned threshold value of queue length for Sensor i

The variables controlled by data cards are β_{4i} and $(n_{min})_i$, $1 \le i \le S_T$.

3.2.2.4 Simulation Output. In addition to the output data available from the programs written during Phase I, the running-average filtering program puts out the following data:

- Time histories of T_i , with start and stop time controllable, for any sensor (4020 plot)
- Histogram of T_i for any sensor (4020 plot)
- Average value of T_i for any sensor



$(n_{min})_i$ = ASSIGNED THRESHOLD VALUE OF QUEUE LENGTH FOR SENSOR i

Fig. 3-9 **Running-Average Filtering Control Function, Queue Length Monitoring System**

3.2.3 Sample Deletion Only

The simulated system is also capable of sample deletion before compression, without any prior filtering. This deletion is such that, for a given sensor, one out of every T samples is retained. The sample deletion is done on an individual sensor basis. The monitored parameter controls T_i , $1 \le i \le S_T$. The quantity T_i is made to vary in such a manner that the resulting sample rate, f_{si}/T_i , varies stepwise linearly with the monitored parameter, where f_{si} = original sample rate for Sensor i.

The control curves and equations for both the compression ratio and the queue length monitoring system are identical to those shown in Figs. 3-8 and 3-9, and presented in Pars. 3.2.2.2 and 3.2.2.3, except that for sample deletion only there is no upper limit on T_i with the queue length monitoring system. Also for sample deletion only, T_i is redefined as the number of samples in the group from which one is retained.

The variables controlled by data cards and the simulation outputs are the same as those presented in Par. 3.2.2.

3.2.4 Additional Features

3.2.4.1 <u>Reconstruction.</u> When samples are deleted before compression, the deleted samples are reinserted into the reconstructed data before the data are plotted or error calculations are made. When deleted samples fall between two end points of a line segment, they are reinserted on that line segment. However, when they fall between two line segments, they are reinserted on a straight line drawn between the closest end points of the two segments. In all the new control systems, errors in the reconstructed data are determined with respect to the original (unfiltered) data.

3.2.4.2 <u>Fixed-Parameter Operation.</u> The programmed filtering and/or sample deletion systems are capable of operating with fixed values of $T_i \neq 1$, or $b_i \neq 0$, $1 \le i \le S_T$, i.e., without feedback.

Section 4
TASK IX, THEORETICAL BUFFER ANALYSIS

4.1 PREDICTIVE QUEUING CONTROL

As presented in the First Quarterly Report (Ref. 1), the sign of the second term of Eq. (4.19b) was in error. This error and those arising from it have been corrected and are presented in Appendix I. This Appendix may be physically substituted for the erroneous para. 4.3 of Ref. 1.

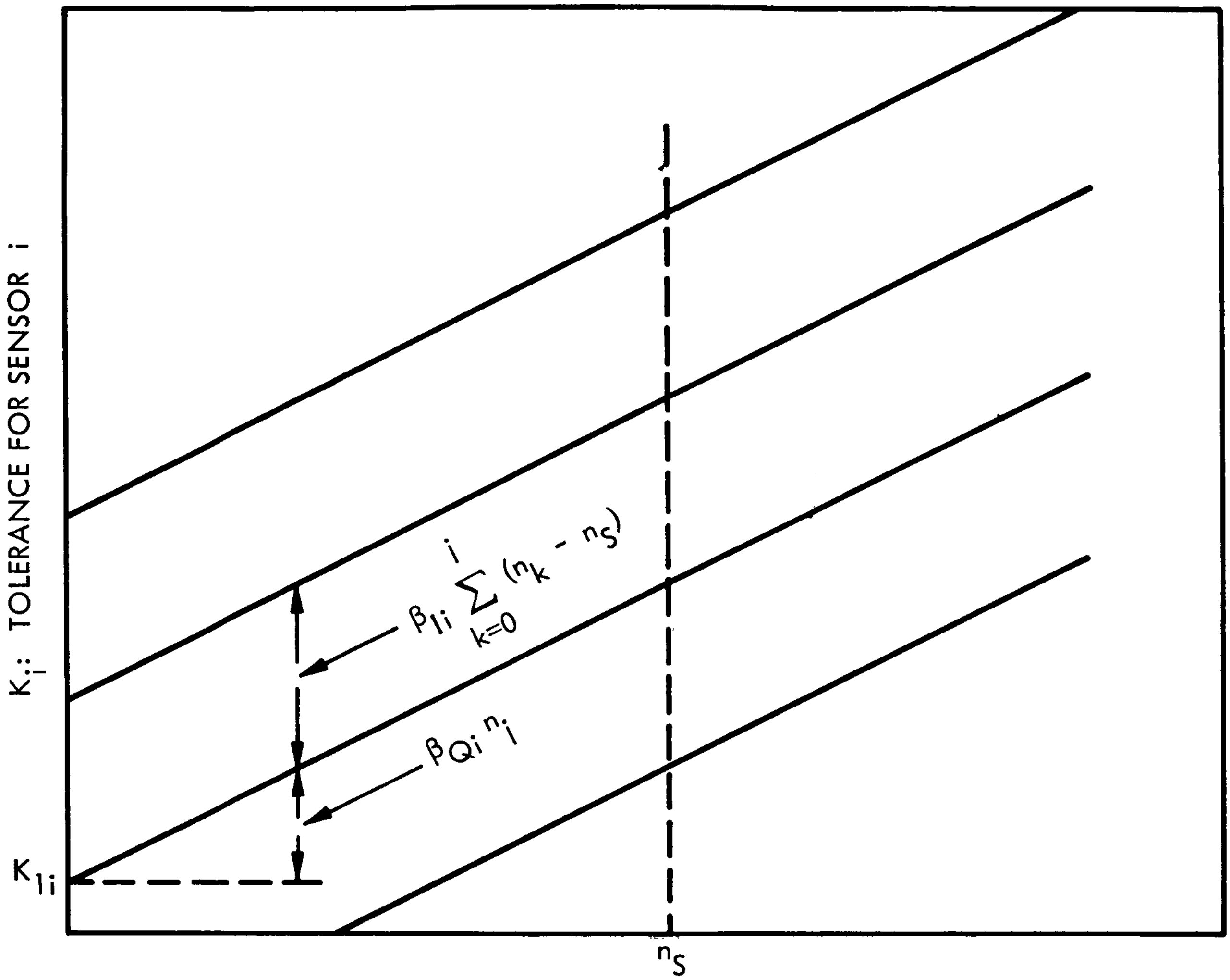
The conclusions which were drawn concerning this erroneous paragraph were not materially affected by this error, and consequently still apply.

4.2 SIMULATION OF COMBINATION QUEUE, INTEGRAL QUEUE FEEDBACK

One of the results of the theoretical buffer analysis (Ref. 1) was that a queuing control using a combination of the queue length and its time integral as the controlling parameter would likely be an effective means of buffer overflow prevention. To further evaluate its potential, the method has been programmed as an additional queuing control subroutine to the multisensor data compression program coded during Phase I. The program has been checked out using S-6 data and appears to be operating properly.

The control curves for the combination queue, integral queue feedback system are shown in Fig. 4-1. Note that the tolerance is controlled on an individual-sensor basis. The programmed control equations are

$$K_i = K_i' = K_{1i} + \beta_{Qi} n_j + \beta_{Ii} \sum_{k=0}^{j} (n_k - n_S) \qquad \text{if } K_i' \geq 0$$



K_{Ii} = INITIAL TOLERANCE FOR SENSOR i

β_{Qi} and β_{Ii} = FEEDBACK CONSTANTS FOR SENSOR i

n_k = QUEUE LENGTH AFTER THE k^{th} INPUT SAMPLE TO THE DATA COMPRESSOR; $k \leq i$

n_S = DESIRED QUIESCENT QUEUE LENGTH

Fig. 4-1 **Control Function for Combination Queue, Integral Queue Feedback System**

$$K_i = O \quad \text{if} \quad K_i' < O$$

where

K_i	= tolerance for Sensor i
K_{1i}	= initial tolerance for Sensor i
β_{Qi} and β_{Ii}	= feedback constants for Sensor i
n_j	= present queue length (after the j^{th} input sample to the data compressor)
n_k	= queue length after the k^{th} input sample to the data compressor; $k \leq j$
n_S	= desired quiescent queue length

The variables controlled by data cards are n_S, K_{1i}, β_{Qi}, and β_{Ii}, $1 \leq i \leq S_T$, where S_T = total number of sensors.

Section 5
TASK X, DATA COMPRESSION MECHANIZATION STUDY

As discussed in para. 1.2.4, Task X has three major parts. These are: (1) a system configuration analysis, (2) a study of secondary data compression, and (3) a transmission link simulation study. Because the outcome of the transmission link simulation tests will influence the conduct of other Phase II studies, activity on Task X during the first six months of Phase II has been confined to this subtask.

The master plan for the transmission link simulation study, discussed in Section 5 of Ref. 1, has been altered slightly during this reporting period in order to conserve computer costs and study time. The planned method of simulating the uncompressed PCM transmission link shown in Fig. 5-2 of Ref. 1 was simplified to that shown in Fig. 5-1 of this report. Instead of adding errors to the original Goddard data tape and generating contaminated TAD and T-3 tapes for each set of noise parameters used, errors will be added directly to the data magnitude words on the TAD tape. The contaminated TAD tape will then be compared directly with the uncontaminated TAD tape with the error analysis program as shown in Fig. 5-2, thus eliminating the necessity for generating contaminated T-3 tapes. The error analysis program will obtain error statistics on a combined sensor basis, as well as for individual sensors. Thus Figs. 5-2 and 5-3 in Ref. 1 are superseded by Figs. 5-1 and 5-2 of this report.

5.1 CPCM BIT STREAM CODING

5.1.1 S-6 Telemetry Data

As the study progressed during the second quarter of Phase II, it was found desirable to modify slightly some of the data formats discussed in para. 5.1 of Ref. 1. These changes are presented in the following paragraphs.

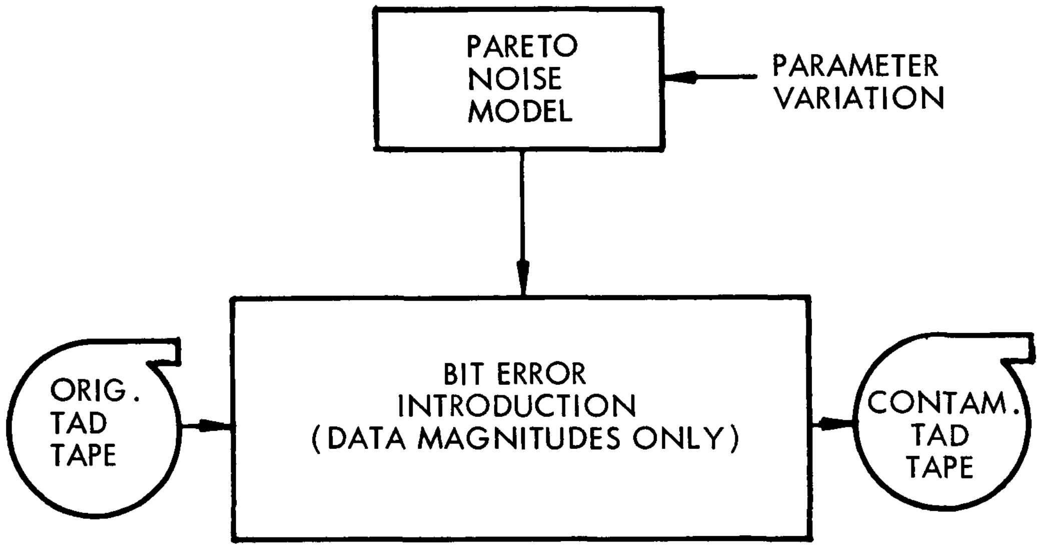
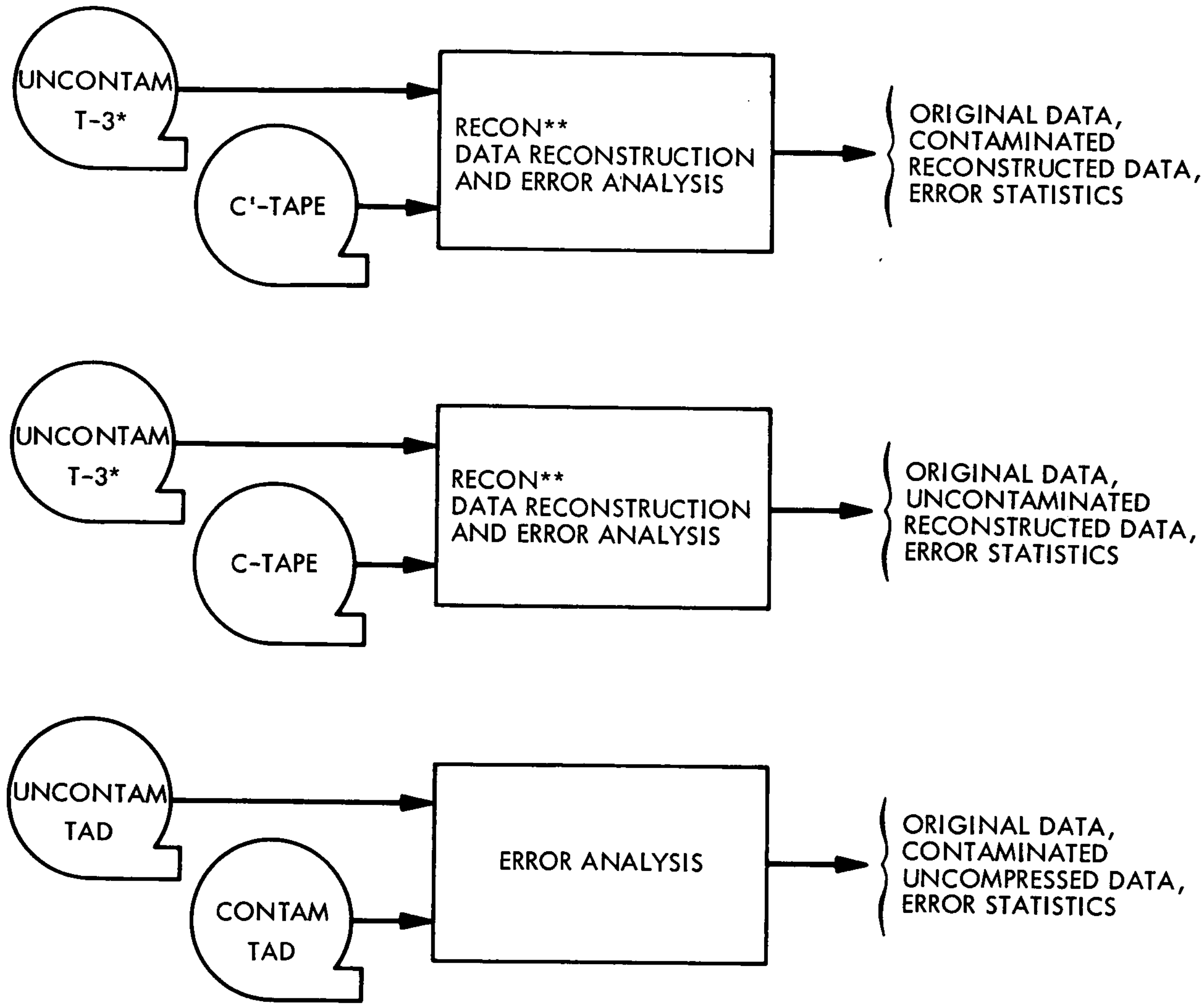


Fig. 5-1 PCM Transmission Link Simulation



NOTES: *OBTAINED FROM ORIGINAL GODDARD TAPE, VIA DECOM AND SAS PROGRAMS

**COMPUTER PROGRAM PREPARED DURING PHASE I (REF. 2, FIG. IV-1)

Fig. 5-2 Data Reconstruction and Error Analysis

5.1.1.1 Amendments to the S-6 Multiplexer Format. As mentioned in Ref. 2, Sensors 17 through 23 were erroneously thought to be cross-strapped. This fact was not taken into account in Ref. 1; thus Fig. 5-4 of Ref. 1 still shows these sensors as cross-strapped. The corrected main frame matrix is shown in Fig. 5-3. Note that there now are 64 sensors instead of 57.

5.1.1.2 Change in the CPCM Format. Because other methods of checking the time word are sufficient to properly decode the CPCM bit stream, the parity check on the time words has been eliminated. The elimination of the two parity bits requires a revision of Table 5-1 and Figs. 5-5, 5-6, and 5-7 of Ref. 1. Table 5-1, Figs. 5-4, 5-5 and 5-6 are the updated versions. Note that the bit stream format for the zero-order interpolator (ZOI) and the zero-order predictor (ZOP) are the same; hence, the letter Z in the table and illustrations stands for either zero-order selector.

Although budget will not allow any further consideration of zero-order selectors in the transmission link simulation study during Phase II, their evaluation would be desirable to serve as a check on the estimates made for this performance criterion in the compression model tradeoff analysis of Phase I (Ref. 2, Par. 6.4.7, and Table 6-4).

Table 5-1

CPCM BIT STREAM RECORD BREAKDOWN FOR S-6 DATA

	FOI	Z
Magnitude bits	864	423
Channel ID bits	192	188
Parity bits	48	47
Flag bits	48	47
Sync bits	24	15
Time and Position bits	46	40
Unused bit(s)	2	5
TOTAL	1,224	765

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	2	4	5	6	7	2	8	9	11	12	2	13	15	16	3	18	2	4	5	6	7	2	10	9	11	12	2	13	14	27	8	43	2	4	5	6	7	2	3	9	11	12	2	13
2	1	2	4	5	6	7	2	8	9	11	12	2	13	15	16	3	19	2	4	5	6	7	2	10	9	11	12	2	13	14	28	8	44	2	4	5	6	7	2	3	9	11	12	2	13
3	1	2	4	5	6	7	2	8	9	11	12	2	13	15	16	3	20	2	4	5	6	7	2	10	9	11	12	2	13	14	29	8	45	2	4	5	6	7	2	3	9	11	12	2	13
4	1	2	4	5	6	7	2	8	9	11	12	2	13	15	16	3	21	2	4	5	6	7	2	10	9	11	12	2	13	14	30	8	46	2	4	5	6	7	2	3	9	11	12	2	13
5	1	2	4	5	6	7	2	8	9	11	12	2	13	15	16	3	22	2	4	5	6	7	2	10	9	11	12	2	13	14	31	8	47	2	4	5	6	7	2	3	9	11	12	2	13
6	1	2	4	5	6	7	2	8	9	11	12	2	13	15	16	3	23	2	4	5	6	7	2	10	9	11	12	2	13	14	32	8	48	2	4	5	6	7	2	3	9	11	12	2	13
7	1	2	4	5	6	7	2	8	9	11	12	2	13	15	16	3	58	2	4	5	6	7	2	10	9	11	12	2	13	14	33	8	49	2	4	5	6	7	2	3	9	11	12	2	13
8	1	2	4	5	6	7	2	8	9	11	12	2	13	15	16	3	59	2	4	5	6	7	2	10	9	11	12	2	13	14	34	8	50	2	4	5	6	7	2	3	9	11	12	2	13
9	1	2	4	5	6	7	2	8	9	11	12	2	13	15	16	3	60	2	4	5	6	7	2	10	9	11	12	2	13	14	35	8	51	2	4	5	6	7	2	3	9	11	12	2	13
10	1	2	4	5	6	7	2	8	9	11	12	2	13	15	16	3	61	2	4	5	6	7	2	10	9	11	12	2	13	14	36	8	52	2	4	5	6	7	2	3	9	11	12	2	13
11	1	2	4	5	6	7	2	8	9	11	12	2	13	15	16	3	62	2	4	5	6	7	2	10	9	11	12	2	13	14	37	8	53	2	4	5	6	7	2	3	9	11	12	2	13
12	1	2	4	5	6	7	2	8	9	11	12	2	13	15	16	3	63	2	4	5	6	7	2	10	9	11	12	2	13	14	38	8	54	2	4	5	6	7	2	3	9	11	12	2	13
13	1	2	4	5	6	7	2	8	9	11	12	2	13	15	16	3	64	2	4	5	6	7	2	10	9	11	12	2	13	14	39	8	55	2	4	5	6	7	2	3	9	11	12	2	13
14	1	2	4	5	6	7	2	8	9	11	12	2	13	15	16	3	24	2	4	5	6	7	2	10	9	11	12	2	13	14	40	8	56	2	4	5	6	7	2	3	9	11	12	2	13
15	1	2	4	5	6	7	2	8	9	11	12	2	13	15	16	3	25	2	4	5	6	7	2	10	9	11	12	2	13	14	41	8	57	2	4	5	6	7	2	3	9	11	12	2	13
16	1	2	4	5	6	7	2	8	9	11	12	2	13	15	16	3	17	2	4	5	6	7	2	10	9	11	12	2	13	14	26	8	42	2	4	5	6	7	2	3	9	11	12	2	13

Fig. 5-3 S-6 Main Frame

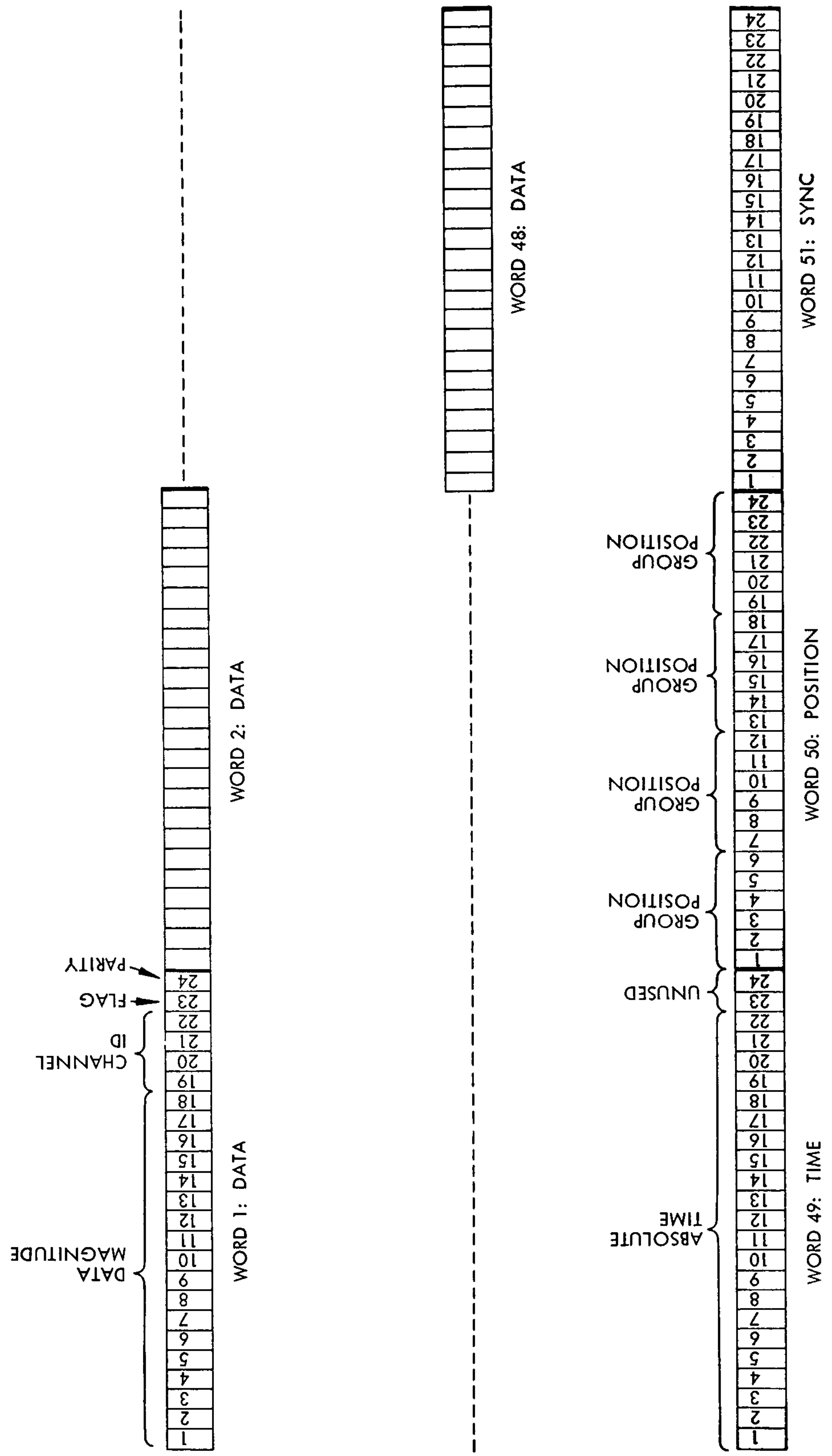


Fig. 5-4 CPCM Bit Stream Record Breakdown for S-6 Data, FOI Selectors

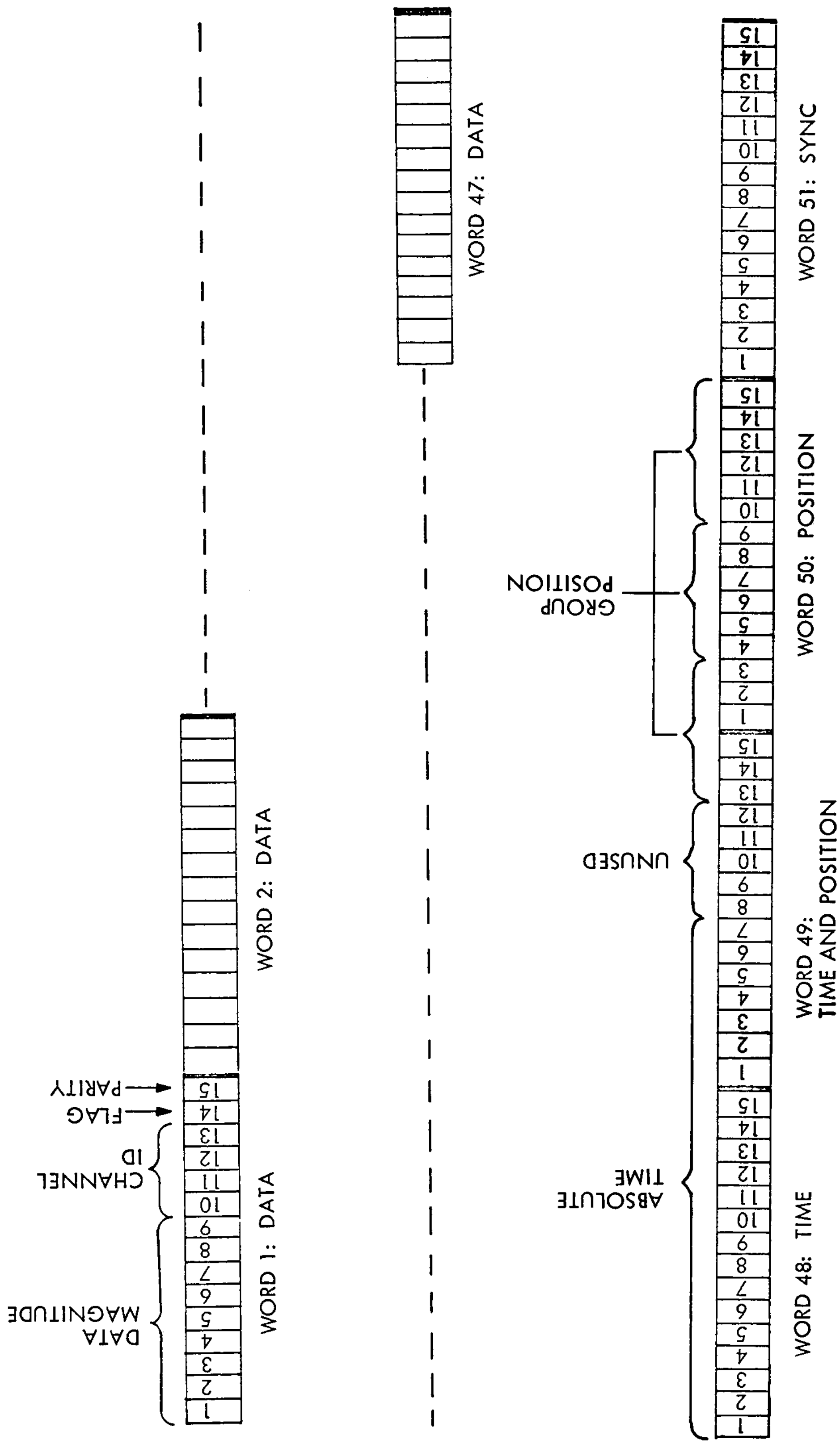


Fig. 5-5 CPCM Bit Stream Record Breakdown for S-6 Data, Z Selectors

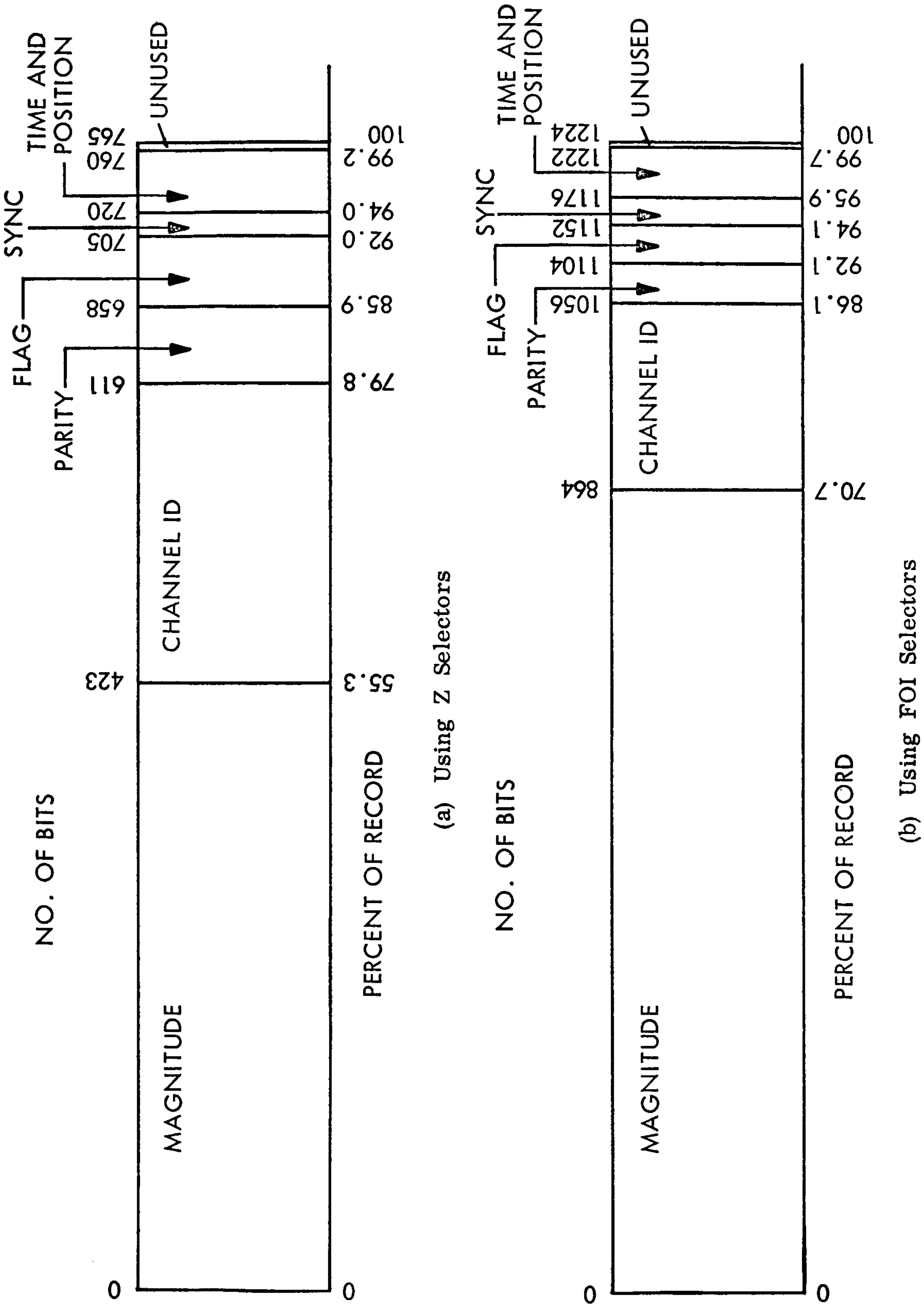


Fig. 5-6 Bit Stream Records for S-6 Data

5.1.2 S-49 Telemetry Data

As in the case of the S-6 CPCM bit stream, the S-49 bit stream must conform to two predetermined data formats. The first one is the multiplexer format of S-49. Figure 5-7 shows one major frame composed of 128 channels, with some of these cross-strapped. Note that channels 97, 98 and 99 are not used; these are subcommutated channels and were omitted at the request of the customer. The second fixed format is the way the compressed data are written on the C-Tape. This constraint was discussed in Ref. 1.

The basic CPCM bit stream format used in this study was discussed in Ref. 1. The coding for the S-49 data is the same as that for the S-6 data, except for the following changes:

- A more accurate time word – The sampling rate on the S-49 is much faster than for the S-6; thus 34 bits instead of 22 bits must be used.
- Number of channels per group – For the S-49, there are 16 channels per group instead of 15.
- The position word – Since there are no submultiplexed channels, each pass through the 128-channel minor frame represents a main frame. Thus there are only 8 possible groups per main frame in which a time word may be placed. This reduces the number of bits for a position word from six to three.

As for the S-6 data, the CPCM bit stream is grouped into records of 51 words allotted differently, depending upon whether a FOI or Z sample selector is used.

5.1.2.1 <u>First-Order Interpolator Selectors.</u> For the FOI selectors, there are forty-eight 24-bit words, one 34-bit time representation (one and one-half words), four 3-bit representations of time position (one-half word) and one 24-bit word for sync. The breakdown of this bit stream is shown following and in Fig. 5-8.

GROUP NO.	CHANNEL ID 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	CHANNEL NO.
	70	71	72	2	3	4	5	62	48	9	65	48	14	33	34	35	SENSOR NO.
2	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	
	66	67	20	21	6	7	8	49	15	62	50	22	23	37	38	39	
3	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	
	73	74	75	10	33	34	35	51	48	9	30	48	62	59	60	61	
4	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	
	24	25	26	22	27	52	11	12	13	17	18	19	33	34	35	62	
5	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	
	76	77	78	53	54	55	56	57	48	9	68	48	14	63	64	36	
6	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	
	1	1	1	1	62	33	34	35	15	30	40	41	42	37	38	39	
7	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	
	X	X	X	31	69	43	62	16	48	9	10	48	58	59	60	61	
8	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	
	30	28	29	32	33	34	35	62	32	32	32	32	44	45	46	47	

Fig. 5-7 S-49 Main Frame

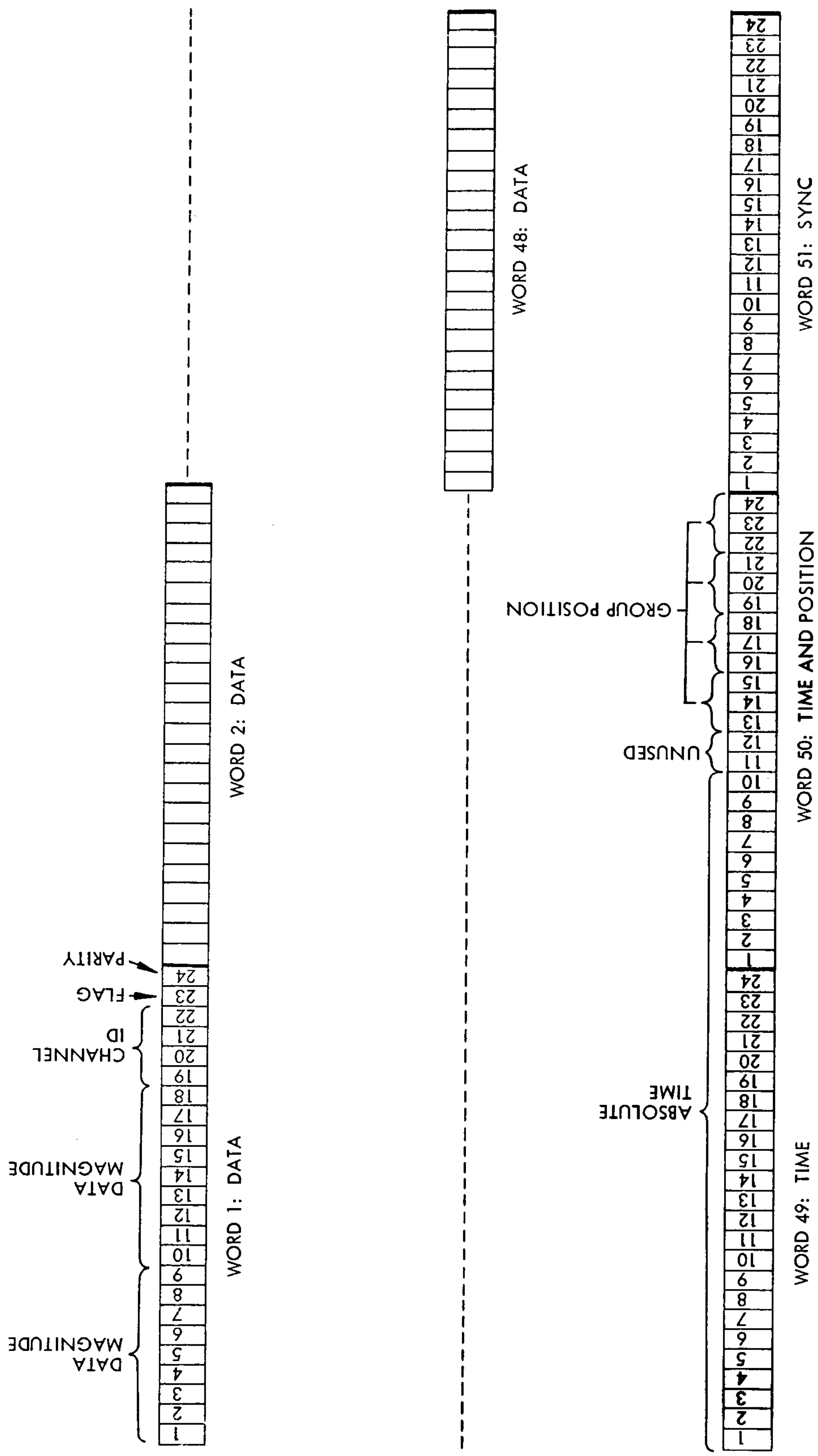


Fig. 5-8 CPCM Bit Stream Record Breakdown for S-49 Data, FOI Selectors

- Data words
  - (a) Eighteen bits for data magnitude
  - (b) Four bits for channel ID
  - (c) One bit for a flag (to indicate a new group of 16)
  - (d) One bit for a parity check on channel ID and flag
- Time word

  Thirty-four bits to pinpoint the absolute time of the out-of-tolerance sample causing the last set of data magnitudes in that bit stream record.
- Position word

  Twelve bits to transmit the exact group of 16 channels in which the time word is located. This word is very important and thus is transmitted four times.
- Sync word

  Twenty-four bits to transmit an artificial sync word composed of all zeros. An error in sync will be caused when some preset number of the zeros are changed to ones. This provides a realistic yet simple method of checking sync.

5.1.2.2 Zero-Order Selectors. As was the case for the S-6 data, the Z selectors again require only 15 bits per word. There are 47 data words, one sync word, two and one-fifth words for time, and four-fifths of a word to represent position. This format completely fills fifty-one 15-bit words; however, there is only room for 33 bits to represent time. The 34th bit is eliminated by transmitting the 33 most significant bits of a 34-bit time word. The accuracy is thus maintained to six decimal places. It is felt that accuracy to this extent is sufficient and the addition of an entire 15-bit word in order to transmit the additional time bit would be unwise. The breakdown of this bit stream is shown following and in Fig. 5-9.

- Data words
  - (a) Nine bits for data magnitude
  - (b) Four bits for channel ID
  - (c) One bit for a flag
  - (d) One bit for a parity check of channel ID and flag.

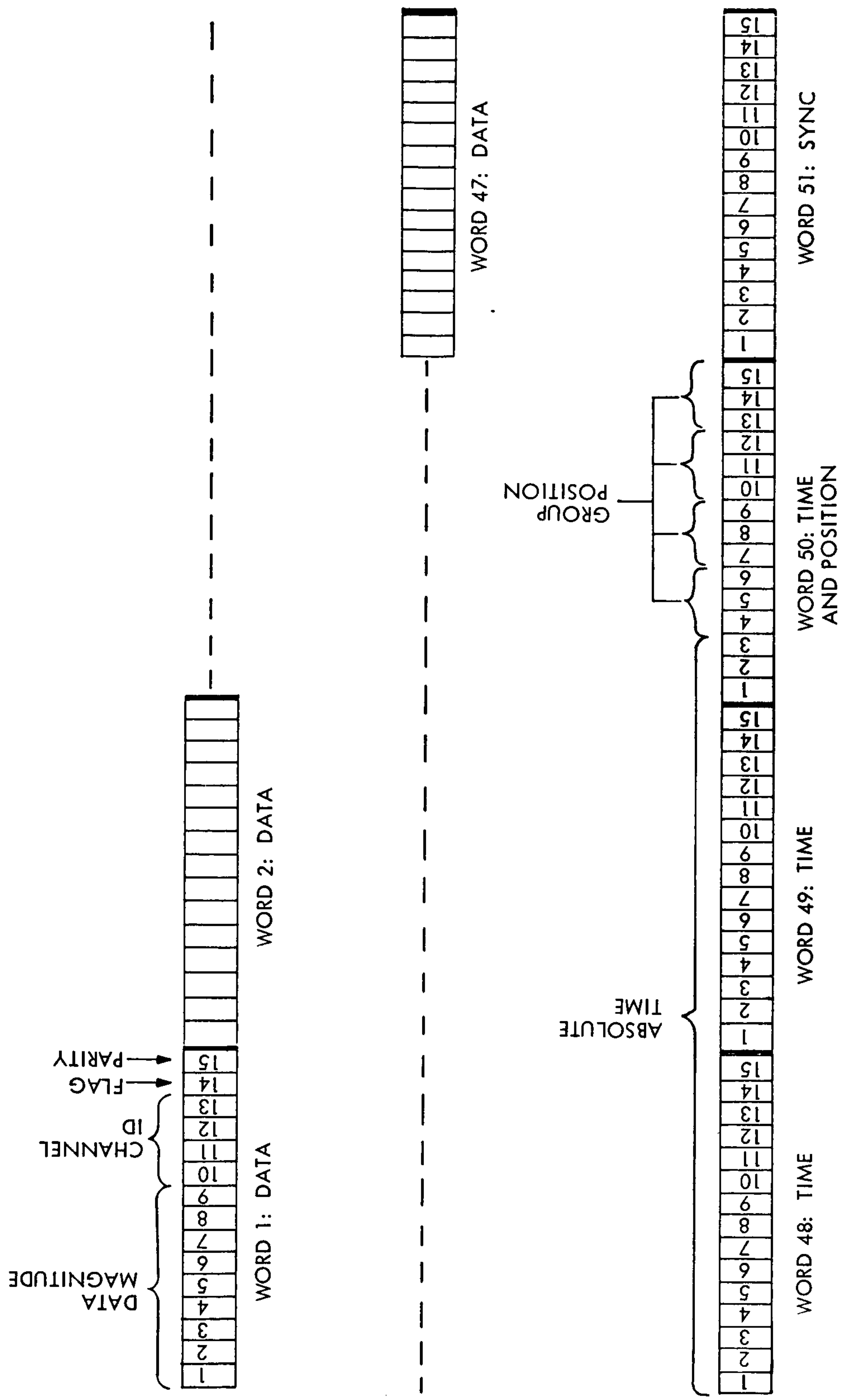


Fig. 5-9 CPCM Bit Stream Record Breakdown for S-49 Data, Z Selectors

- Time word
  Thirty-three bits (the 33 most significant bits from a 34-bit time word) with two complete words plus three bits in a third word.
- Position word
  Twelve bits to transmit the exact group of 16 channels in which the time word is located. Due to the importance of this word, it is transmitted four times.
- Sync word
  The sync word is the same as for the FOI case.

Table 5-2 shows the breakdown of the FOI and Z bit stream records on a bit basis.

Table 5-2

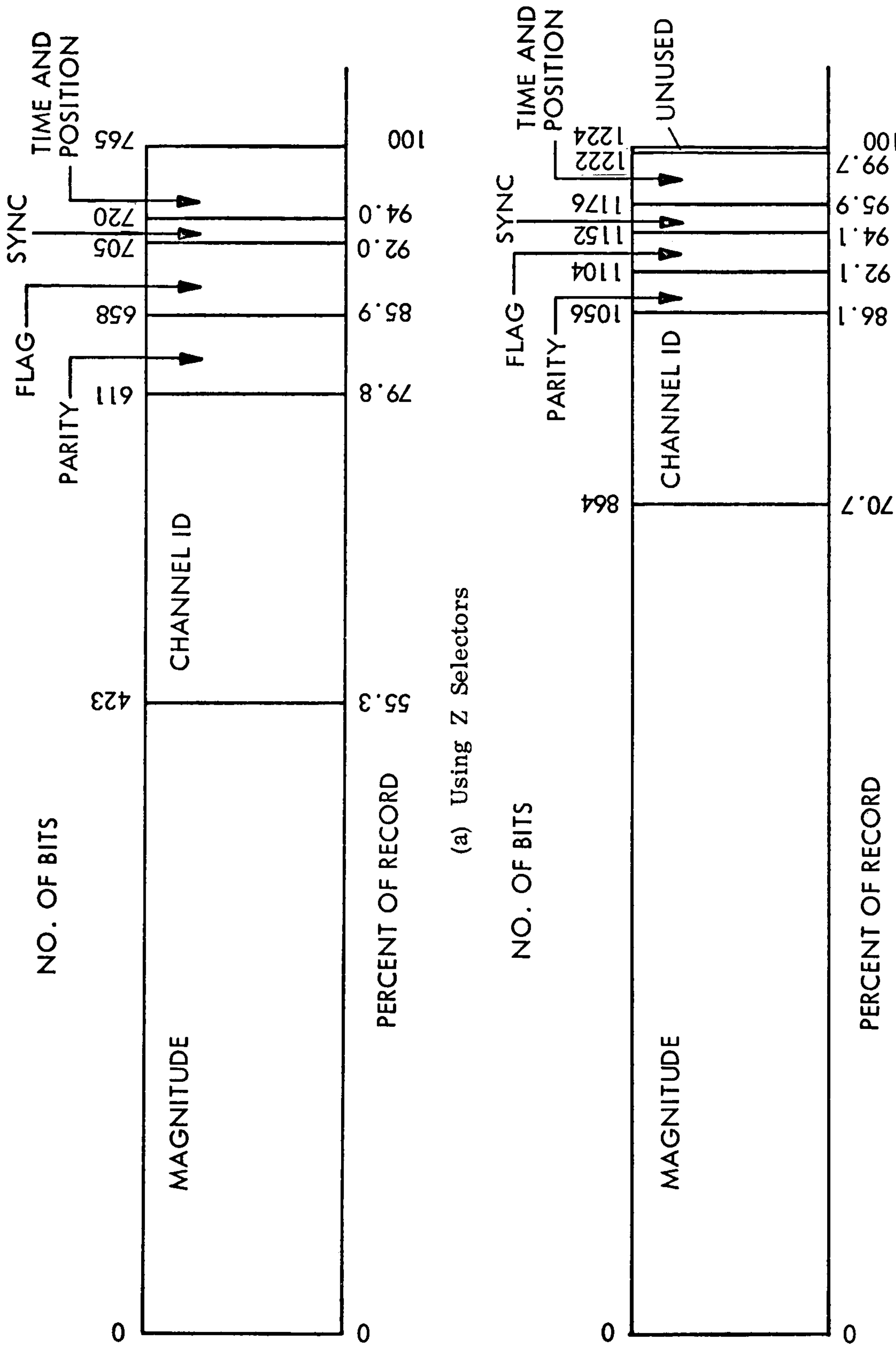
CPCM BIT STREAM RECORD BREAKDOWN FOR S-49 DATA

	FOI	Z
Magnitude bits	864	423
Channel ID bits	192	188
Parity bits	48	47
Flag bits	48	47
Sync bits	24	15
Time and Position bits	46	45
Unused bit(s)	2	0
TOTAL	1,224	765

Figure 5-10 shows the relative proportions of bit utilization in the bit stream records for Z and FOI selectors. Note that as for the S-6 data, the bits representing data mangitude comprise more than 50 percent of a record.

5.1.3 Bit Requirements and Group Size

Experimental results presented in Ref. 1 demonstrated that using groups of 15 channels instead of 45 channels was a good choice for the S-6 data and the FOIDIS selector.



(a) Using Z Selectors

(b) Using FOI Selectors

Fig. 5-10 Bit Stream Records for S-49 Data

The results following present the same information for the S-6 data using the ZOP selector. Table 5-3 compares the results of processing 3354 line segments using groups of 15 channels and groups of 45 channels.

Table 5-3

CPCM BIT USAGE REQUIREMENTS FOR TWO DIFFERENT GROUP SIZES

	Groups of 15	Groups of 45
Number of bit stream records	72-6/51	71-25/51
Number of groups involved	840	280
Number of empty groups (requiring dummy word transmission)	32	0
Number of real line segments encoded	3,354	3,354
Total number of line segments encoded including dummy words	3,386	3,354
Total number of bits needed to code including time and sync	55,170	58,276
Total number of uncompressed bits including sync. (NOTE: no time was transmitted)	120,960	120,960
Bit compression ratio	2.2	2.08

A comparison of the two bit-stream compression ratios shows that coding in groups of 15 channels is more efficient than coding in groups of 45 channels. Figure 5-11 shows what happens as the number of empty groups of 15 varies, assuming there are no empty groups of 45. (The dotted line indicates the behavior of the curve when empty groups of 45 are taken into account.) For this example, 3.8 percent of the groups are empty; and for groups of 45 to be more efficient, 27 percent of all the groups must be empty.

Another interesting parameter is the ratio between the number of line segments coded (including dummy words) and the number of groups of 15 represented. The ratio is 3,354/840 or 3.88. This means that on the average there are 3.88 words per group of 15. This indicates that coding in groups smaller than 15 might be advantageous for the

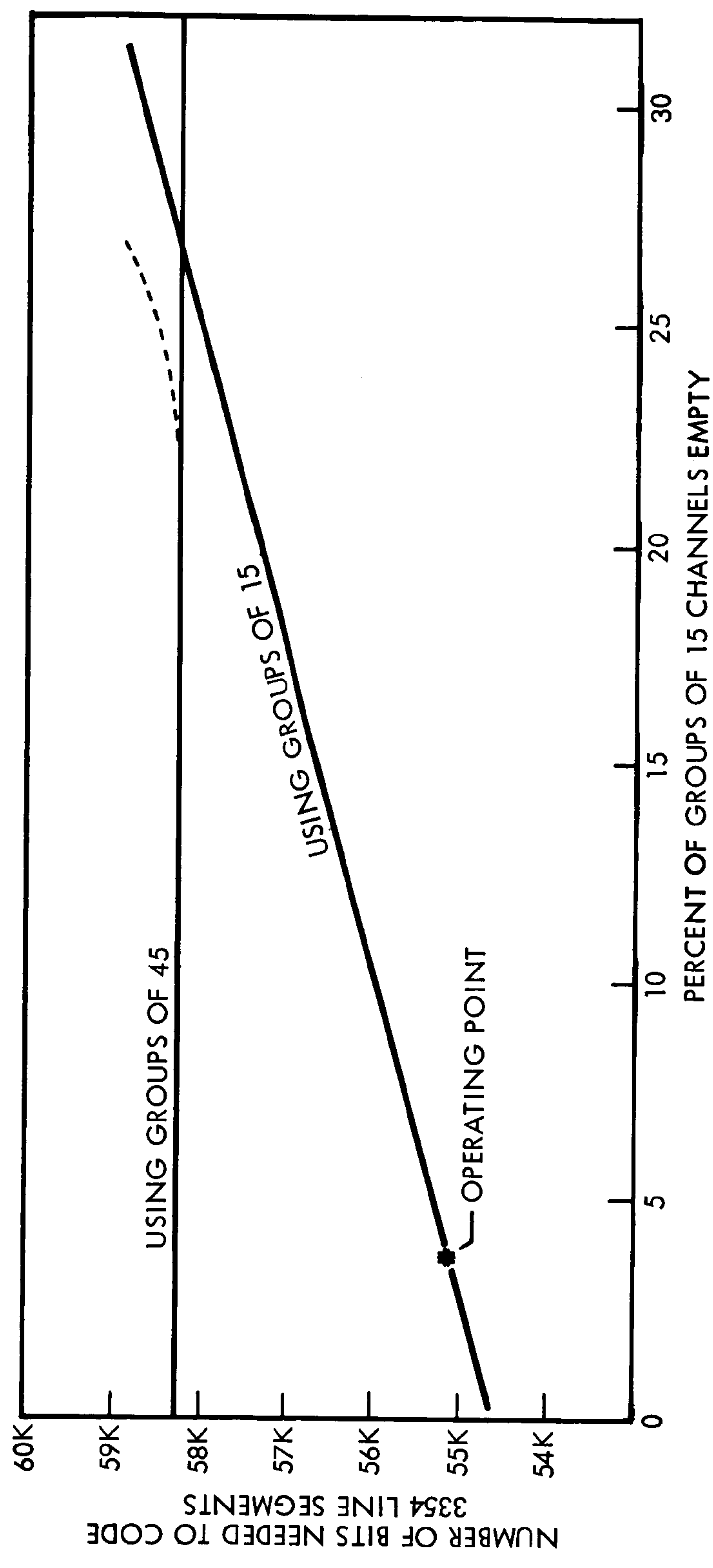


Fig. 5-11 **CPCM Bit Usage Requirements for Two Different Group Sizes**

zero-order predictor on S-6 data. However, the next step would be to code in groups of nine, and that would require just as many bits for coding the channel ID as would groups of 15. It would be necessary to drop all the way down to groups of five in order to become more efficient. Even this may be desirable since on the average there are more than three words per group of 15. In future transmission link noise simulation studies, before the CPCM bit stream format for the zero-order selector is finalized, further consideration should be given to groups of smaller size than 15.

5.1.4 Coding Problems

The biggest problem in the bit error study to this point has been coding the bit stream from the C-Tape. The procedure is quite simple, yet timing inconsistencies on the C-Tapes cause problems, which are pointed out in this paragraph. Following are a few sets of data points from a FOIDIS, S-6, C-Tape.

Line Segment No.	Sensor No.	Beginning Time	Beginning Magnitude	Ending Time	Ending Magnitude
1	2	62.000666	150.0	62.018442	165.3
2	2	62.023997	183.0	62.030663	226.0
3	2	62.036218	285.0	62.041774	371.0
4	2	62.045110	484.0	62.050665	1.0
5	7	62.017331	209.0	62.049554	395.9
6	2	62.056220	1.0	62.086219	1.0
7	7	62.067331	472.0	62.085108	1.0
8	2	62.091774	381.0	62.095111	155.0
9	11	62.003999	165.0	62.103999	197.0
10	2	62.100666	155.0	62.130664	183.2
11	8	62.001777	30.0	62.151777	30.0
12	2	62.136219	194.0	62.173998	172.9
13	12	62.005110	273.0	62.172887	286.0

The initial sensor number and its position in the main frame (Fig. 5-3), are used to start the bit stream; hence, these values are obtained from the tape printout. The position of the second sensor and each subsequent sensor is found in the following manner. The difference between the ending time of a given line segment and the ending time of the previous line segment is noted. (The position in the main frame where this previous end point is located has been stored.) The difference in time is multiplied by the nominal sampling rate (900 samples per second for S-6 data) and the number of sample positions between the end points of those two line segments is found. This number, along with the position of the end point of the previous line segment, allows for the position in the main frame of the end point of the present line segment to be found.

The difficulty in this procedure lies in the fact that the actual sample periods deviate by varying amounts from the nominal period. Because this jitter causes the coding program often to position itself in the wrong square of the main frame, the program must initiate a search procedure to find the proper sensor number. The problem, which appears to be even more serious with the S-49 data, could be solved by adding readings from an input sample counter to the C-Tape, but such a modification is beyond the current scope of the study.

Note that this timing difficulty stems from efforts to simulate, using existing formats, a CPCM transmission link; if, as in an actual system, the bit stream could be coded as the data were compressed, there would of course be no problem.

At this writing, it is believed that the "search procedure" approach to the problem will be successful, at least in the case of the S-6 data, and that valid CPCM bit stream tapes will thus be generated.

5.2 OTHER COMPUTER PROGRAMS

During the second quarter, all the computer programs needed to perform the error analysis on S-6 data were completed and debugged. Three of these routines required revision to handle S-49 data. These changes have been programmed, and the routines are currently being debugged.

5.2.1 Error Addition to Original Data

This program is called S6EROR and adds Pareto model errors to either S-6 or S-49 data. This program adds bit errors to the data magnitude words of the TAD tape, which has the same format regardless of the satellite from which it is derived. The S6EROR program is thus compatible with all uncompressed data formats.

5.2.2 Error Analysis of Uncompressed Data

This program, called TADPLT, has a number of capabilities which are listed below.

- Plots individual sensor data from a single TAD tape.
- Compares contaminated and uncontaminated TAD tapes and generates overall rms error statistics on a time-selective basis.
- Compares contaminated and uncontaminated TAD tapes on an individual-sensor basis and generates rms errors statistics for each sensor.
- Plots uncontaminated and contaminated data for up to three sensors; this may accompany the generation of error statistics.

All the above capabilities are on a time-selective basis.

The advantage of this program is that individual sensors may be plotted without the generation of T-3 tapes. This provides an economical method of viewing new data before particular sensors are chosen for study. TADPLT is thus a help to the entire telemetry study, besides being useful to Task X in particular.

5.2.3 Judgment of Contaminated Data

The problem of determining a reasonable, yet sufficient, judgment routine is indeed difficult. There are a number of bit error judgment items on CPCM that are worthy of note. Various items could be used, depending upon the sophistication and accuracy desired in decoding the contaminated data. A list of possible judgment items is shown below:

1. Check for the agreement of the time words with the corresponding group position words, and for increasing time. This check allows for the detection of incorrect timing, which would cause faulty decoding of the bit stream record associated with that time word.
2. Parity check of channel ID and flag. This check provides for the detection of individual word errors within a record.
3. Check for monotonically increasing channel ID between flag bits. This check discovers some errors in channel ID and flag that may escape the parity check.
4. Check individual sensors for increasing time. This check, which eliminates the possibility of double-valued data from a particular sensor, is separate and distinct from the time checks at the end of each bit stream record (Item 1).

These four checks would provide the maximum amount of error detection with the present level of redundancy. Obviously, with an increase in redundancy further detection and error correction could be added; however, this will not be studied under this phase of the study.

Since one of the study objectives is to determine how badly bit errors hurt compressed data, a minimum amount of judging will be used at first. As time allows, more elegant checks may be used. To begin with, only Checks 1 and 4 will be employed; thus only time errors will be detected. Check 1 will assure that errors in one bit stream record do not cause errors in subsequent records as they are decoded. If this check were not made, the first time a time word was in error all the rest of the decoding would also be in error. Check 4 will make sure that, for the purpose of reconstruction, time is always increasing. Time reversals not only look bad, but they also could cause problems with the routines that plot the reconstructed data. The routine using these two checks is called JUDGEA. These checks merely detect errors and eliminate them; there is no attempt at error correction.

Note that there is never any check on data magnitude. There are two reasons for this. First, since no attempt is made to correct errors there is no real need to know about an error in magnitude. Second, the normal way to measure system error is to find the rms error between the original data and the reconstructed data; to remain consistent with this philosophy, the judging routines in general, and JUDGEA in particular, attempt to "translate" all errors into magnitude errors. This means the following:

- Errors in channel ID will cause data magnitudes for that line segment to be assigned to an incorrect sensor, showing up as errors in magnitude.
- Errors in a flag bit cause subsequent groups of 15 (or 16) channels to have incorrect sensor assignments. This means that all the data through the end of the record being decoded will have incorrect sensor IDs (unless there was an offsetting error in a flag bit before the end of the record). The time and position words at the end of each record serve as a break and set the decoding process right again. As an example of this type of error, assume a location in the second group of 15 channels in row 1 of the matrix shown on Fig. 5-3. If the flag bit signifying the beginning of that group were missing, the decoding program would assign sensor values of 1, 2, 4, 5, 6 . . . 16 to the channels instead of 3, 18, 2, 4, 5 . . . 14. If an extra flag bit were present the sensor values in the third group would be assigned.

There is a method of sensor assignment which overcomes this problem in particular cases. If the main frame shown in Fig. 5-3 was broken into two groups of 16 channels followed by a group of 13 channels more efficient coding is produced. Now the second position in each group is Sensor 2, the third position is Sensor 4, and so on for a total of eleven positions in each of the three groups. Thus an error in the flag bit would not cause the incorrect assigning of data magnitudes in 33 of the 45 minor frame positions. However, there would be an error in time proportional to the number of groups that were skipped, even in the 33 cases where data magnitudes had been assigned to correct sensors.

Although this method seems advantageous, it was not used because it was desired to divide the 45 S-6 channels into equal groups. It was not used in the case of the S-49 data, since there is no correlation between sensors and minor frame channels. However, a sensor assignment of this type, if taken into consideration when the multiplexer format was arranged, would surely reduce the effect of flag bit errors.

5.3 BIT ERROR MODEL

The bit error model to be used, and its method of generation, is described in Ref. 1. The model will follow a Pareto distribution. During this reporting period, the method and range of noise parameter variation was established.

There are four parameters that determine the Pareto error model; these include the smallest allowable gap, the largest allowable gap, the average gap, and the clustering factor. Since we are dealing with a discrete system, it is obvious that the smallest allowable gap should be unity. An approximation of the average gap may be expressed as (Ref. 3)

$$g_{av} = \frac{L^{(1-\alpha)}}{1 - \alpha}$$

where

g_{av} = the average gap (1/error rate)
α = the clustering factor
L = the largest allowable gap

This formula provides a way to gain information about the other three Pareto model parameters. Figure 5-12 shows a plot of average gap versus largest possible error for various values of α. Sussman (Ref. 3) indicates that the average gap generally varies from 1,000 to 1,000,000, with most cases having average gaps between 10,000 and 100,000. He states further that the value of α ranges from 0.1 to 0.4. Due to considerations of statistical sample size and computer time, the average gaps to be used will be at the low end of the range mentioned. To use the widest range of α and

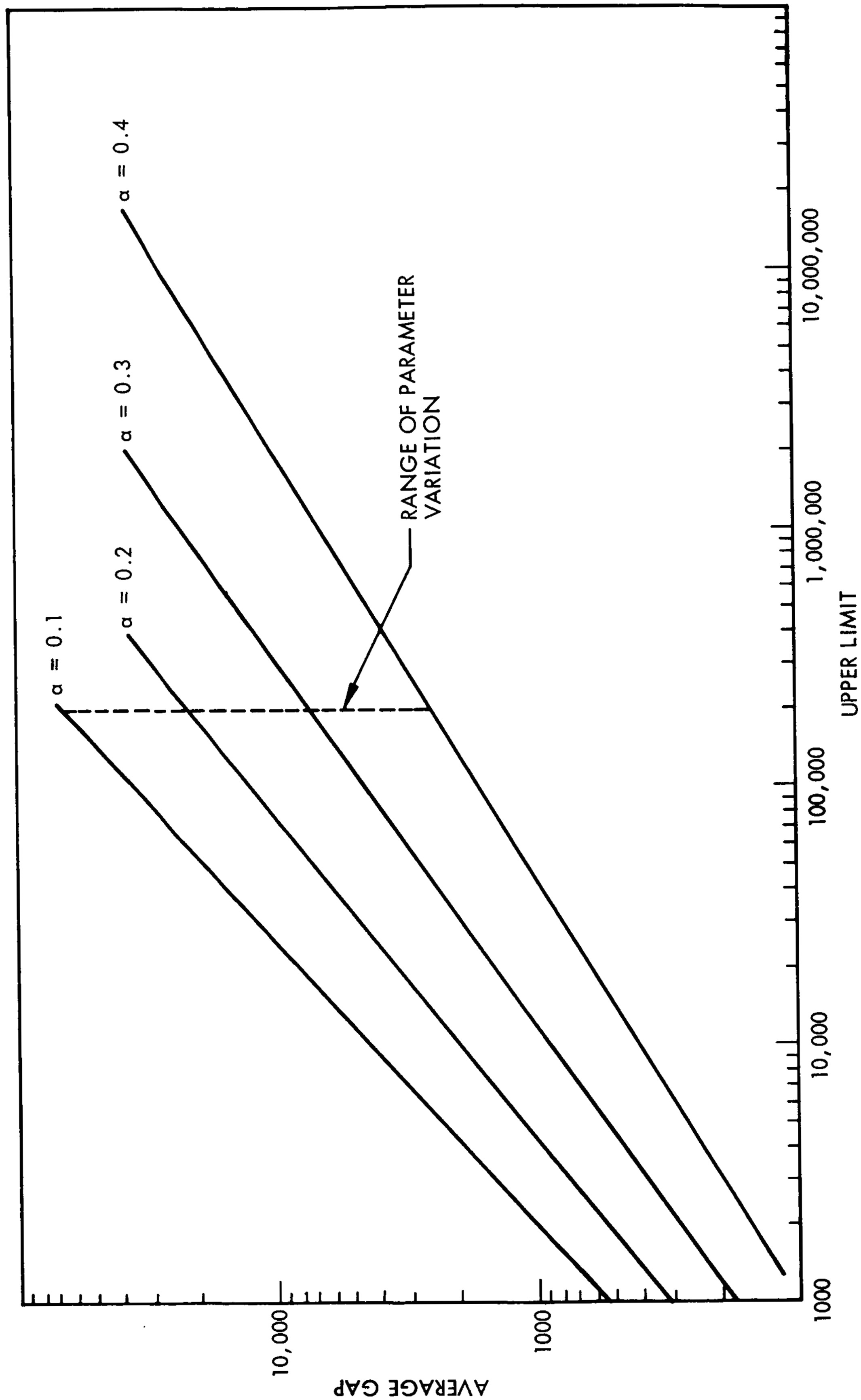


Fig. 5-12 Pareto Noise Model Parameters

average gaps, in view of the study limitations, the upper limit will be held at 200,000 and α varied from 0.1 to 0.4. This will produce average gaps ranging from 2,500 to 66,000, which covers much of the area discussed by Sussman. This range is shown as a dotted line in Fig. 5-12. The number of experiments performed using parameter values from this line will depend upon available time and computer allotment.

5.4 WORK COMPLETED

The following is a brief review of the work completed on Task X during the reporting period.

- The bit stream decoding aid JUDGEA has been coded in Fortran IV and debugged.
- The bit error addition programs for both CPCM and uncompressed data have been coded in Fortran IV and debugged.
- Various Pareto noise model parameters have been studied, and a region of operation has been set.
- The transformation of S-6 programs into programs which handle S-49 data has been completed.
- Production runs involving this task and the S-6 data have begun.

Section 6
NEW TECHNOLOGY

No new concepts or techniques were developed during the reporting interval.

Section 7
PROGRAM FOR NEXT REPORTING INTERVAL

The next interval will involve work on Tasks VII through XIII, as follows.

7.1 TASK VII, S-49 COMPRESSION MODEL ANALYSIS

During the next reporting interval, the sample selector and queuing control evaluations, using data from the S-49, OGO-A satellite, will be continued. Computer run data will be continued. Computer run data will be obtained concerning average compression ratios, queue behavior, and reconstructed data quality, and analyses of these results will be made.

7.2 TASK VIII, NEW DATA COMPRESSION MODEL EVALUATION

7.2.1 New Selector Evaluation

The computer evaluation of the exponential interpolator, short-term frequency component selector, revised bit plane encoding technique, and broadside data compression will continue during the next reporting interval. Average compression ratios and error information will be derived and analyzed.

7.2.2 New Queuing Control Evaluation

Production computer runs on the queuing control test model, employing adaptive filtering and sample deletion, will begin during the next interval. Queue behavior and reconstructed data quality will be observed and compared for the different control systems and parameters used.

7.3 TASK IX, THEORETICAL BUFFER ANALYSIS

Computer-evaluation runs on the simulated combination queue, integral queue feedback system will be started. A preliminary analysis of the results will be made.

7.4 TASK X, DATA COMPRESSION MECHANIZATION STUDY

Production test runs on the transmission link simulation study will be continued during the next period. Experimental data on the relative effect of ground transmission link noise on compressed and uncompressed telemetry data will be obtained.

The investigation of secondary data compression (Par. 1.2.4.2) will begin.

7.5 TASK XI, THEORETICAL TRANSMISSION ERROR ANALYSIS

The portion of this task which analyzes the experimental data obtained in the transmission link noise simulation tests of Task X (Par. 1.2.5) will begin during the next reporting interval.

7.6 TASK XII, REAL TIME COMPUTER ANALYSIS

During the next reporting period, the Goddard Space Flight Center requirements for real-time data analysis will be determined. Functional flow patterns will be established, and instruction counts will be estimated for accomplishing each required function.

7.7 TASK XIII, DATA PROCESSING COMPUTER ANALYSIS

Goddard Space Flight Center workloads and data processing requirements will be determined during the next interval. The definition of functional flow patterns will begin.

Section 8
CONCLUSIONS AND RECOMMENDATIONS

With the exception of Task IX, none of the tasks of Phase II have yet progressed to the point that conclusions can be made regarding the main points of the study. Conclusions and recommendations concerning the theoretical portion of Task IX are presented in Ref. 1. The following paragraphs contain a number of observations on the remainder of the work performed to date.

8.1 TASK VII, S-49 COMPRESSION MODEL ANALYSIS

From the results of the S-49 exploratory multisensor compression and buffer simulation computer run described in Section 2, and those of similar runs on S-6 data, a comparison can be made of the overall compressive characteristics and buffer input statistics of the data from the two satellites. Reference 2 shows that the overall compressibility of the S-6 data was relatively low; in a typical compression run, the FOIDIS selector obtained a combined sample compression ratio of 4.1. The S-49 data, on the other hand, showed a combined compression ratio of 13.4 with the same selector, and with comparable resolution restrictions. This relatively high compression efficiency stems, of course, from the slowly varying nature of a large number of S-49 sensor waveforms.

The statistical nature of the S-6 data input to the buffer exhibited, in the absence of transmitted bit errors, a marked stationarity. The statistics observed from the S-49 run show that the buffer input rate may not be as steady for these data. This may be caused in part by bit errors in the S-49 data, although the data specimen used in the run is known to be 99.89 percent error free. The apparent dependence, in part, of the buffer input rate on the satellite roll position suggests that if the OGO-A satellite had been stabilized, the buffer input characteristics may have been more stationary. At any rate, this non-stationarity will provide a good environment for the empirical evaluation of queuing control techniques.

8.2 TASK X, DATA COMPRESSION MECHANIZATION STUDY

Reference 1 showed that bit-stream coding in groups of 15 channels was an optimum method of channel identification for FOI data. On the other hand. Par. 5.1.3 indicates that coding in much smaller groups may be reasonable for Z selectors. Therefore, in future transmission link noise simulation studies before the CPCM bit-stream format for the zero-order selector is finalized, further consideration should be given to groups of smaller size than 15.

A special-case method of reducing the effect of bit errors on sensor identification was presented in Par. 5.2.3. This system is feasible only in certain systems, but its use in these systems would greatly reduce the effect of bit errors on sensor identification.

Section 9
BIBLIOGRAPHY

1. Lockheed Missiles & Space Company, First Quarterly Report for PCM Telemetry Data Compression Study, Phase II, by Bechtold, Bjorn, and Medlin, M-62-66-1, Sunnyvale, California, Jan 1966

2. -----, Final Report for PCM Telemetry Data Compression Study, Phase I, by Bechtold, Medlin, and Weber, M-62-65-2, Sunnyvale, California, Oct 1965

3. ADCOM, Inc., Study Evaluation, and Analysis of Unified S-Band System for Apollo Ground Network, Fourth Quarterly Progress Report: 1 April 1965—30 June 1965, Contract No. NAS 5-9702, Cambridge, Mass., 1965

Appendix I
PREDICTIVE QUEUING CONTROL

The following material supersedes that presented in Par. 4.3 of Ref. 1. To facilitiate its physical replacement, pp. 4-53 and 4-60 of Ref. 1 are also reproduced.

Appendix II
SHORT-TERM FREQUENCY COMPONENT SELECTOR: PRINCIPLE OF OPERATION

The operation of the short-term frequency component selector, which was originally described in Ref. 1, is repeated below for the convenience of the reader.

Using the short-term frequency component selector amounts to making a short-term Fourier analysis of a portion of data from each sensor, and retaining or deleting samples on the basis of this analysis. This selector, although providing no peak error guarantee, tends to maintain the ratio of short-term sampling rate to short-term information bandwidth constant. The mathematical foundation upon which this selector's operation is based is presented in the following paragraph.

The power density, $|Y_s(f)|^2$, of the $2N + 1$ data samples centered about the i^{th} data sample is

$$|Y_s(f)|^2 = \left[\sum_{n=-N}^{N} Y_{i+n} \cos(2\pi m f T)\right]^2 + \left[\sum_{n=-N}^{N} Y_{i+n} (2\pi m f T)\right]^2 \qquad \text{(II. 1a)}$$

where Y_{i+n} is the $i + n^{th}$ data sample value and T is the data sample period. Because the number of observed data samples $2N + 1$ is finite and the corresponding observation interval $2NT$ is also finite, the frequency resolution obtainable from $2N + 1$ samples of data is $1/2NT$. The power density at frequencies 0, $1/2NT$, $1/NT$ $3/2NT, \ldots m/2NT \ldots 1/2NT$ is

$$|Y_s(m/2NT)|^2 = \left[\sum_{n=-N}^{N} Y_{i+n} \cos \frac{\pi n m}{N}\right]^2 + \left[\sum_{n=-N}^{N} Y_{i+n} \sin \frac{\pi n m}{N}\right]^2 \qquad \text{(II. 1b)}$$

for $m = 0, 1, 2, \ldots N$. The sample selection process consists of measuring cumulative power densities from frequency 0 to $M/2NT$, normalizing this cumulative power density in accordance with the total power in the $2N + 1$ data samples, and making a decision on sample retention based on this normalized cumulative power density. An analytical expression for the normalized cumulative power density, $C_N(M)$, from frequency 0 to $M/2NT$ is

$$C_N(M) = \frac{\sum_{m=o}^{M} |Y_s(m/2NT)|^2}{\sum_{m=o}^{N} |Y_s(m/2NT)|^2} \tag{II. 2}$$

The computer program for this selector is based on Eqs. (II. 1) and (II. 2). The program will allow compression by this selector of a single sensor of Goddard satellite data.

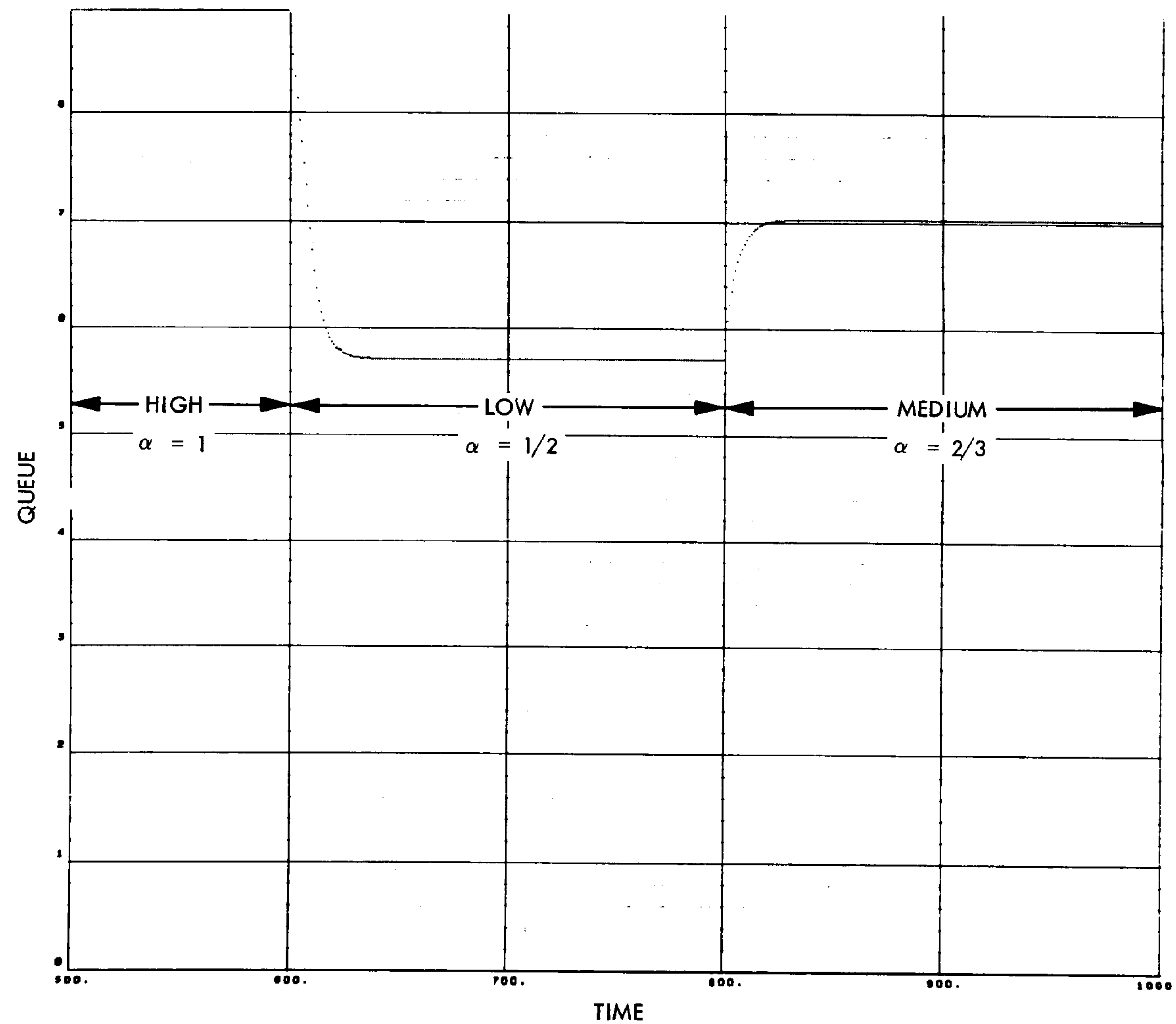


Fig. 4-11d Queue as a Function of Time, Exponential Queue Feedback, Example 2 (Cont'd)

4.3 PREDICTIVE QUEUING CONTROL

Most previous methods of adaptive aperture queuing control have been based on adjustment of tolerance in accordance with some fixed, predetermined function of buffer history. However, a queuing control technique developed during this study period and discussed in the following paragraphs, is based upon adjustment of tolerance in accordance with a variable, non-predetermined function of buffer history. Because of the variable control function, this predictive queuing control system is more flexible than a predetermined queuing control system. Consequently, predictive queuing control may overcome some of the problems associated with determined queuing control.

4.3.1 Description

Let the equation describing the linearized data compression model be

$$Q_m = Q_{m-1} + M\left(1 - \frac{K_m}{\alpha_m} - \frac{1}{R}\right) \tag{4.15a}$$

or equivalently

$$Q_m = Q_{m-1} + \Delta Q_m \tag{4.15b}$$

where Q_m is the result of the m^{th} measurement of buffer queue, K_m is the tolerance used during the m^{th} measurement period, $1/R$ is the ratio of buffer readout rate to input data sample rate, and ΔQ_m is the change in buffer queue from the $(m-1)^{st}$ to the m^{th} measurement. This measurement interval corresponds to M input data sample intervals. [Equation (4.15) is the result of generalizing Eq. (4.5) from a measurement interval of one sample interval to a measurement interval of M sample intervals.]

Assume that the estimate of α_{m+1}, denoted by $\hat{\alpha}_{m+1}$, is simply α_m if the data behaves according to Eq. (4.15). Then $\hat{\alpha}_{m+1}$ can be calculated from Eq. (4.15) as follows:

$$\hat{\alpha}_{m+1} = \alpha_m = K_m \left[\frac{M}{M\left(1 - \frac{1}{R}\right) - \Delta Q_m} \right] \tag{4.16}$$

If the data behavior is truly in accordance with Eq. (4.15), and if $\hat{\alpha}_{m+1}$ is a correct estimate of α_{m+1}, K_{m+1} can be chosen to give any projected changes in buffer queue $\hat{\Delta Q}_{m+1}$ during the $(m+1)^{st}$ measurement interval. Using Eqs. (4.15) and (4.16), the following equations are obtained:

$$\hat{\Delta Q}_{m+1} = M\left(1 - \frac{K_{m+1}}{\hat{\alpha}_{m+1}} - \frac{1}{R}\right) \tag{4.17a}$$

so that

$$\left(\frac{K_{m+1}}{K_m}\right) = \frac{\left[M\left(1 - \frac{1}{R}\right) - \hat{\Delta Q}_{m+1}\right]}{\left[M\left(1 - \frac{1}{R}\right) - \Delta Q_m\right]} \tag{4.17b}$$

More specifically, one might choose K_{m+1} so that $\hat{\Delta Q}_{m+1}$ is zero. In this case. Eq. (4.17b) becomes

$$\left(\frac{K_{m+1}}{K_m}\right) = \frac{M\left(1 - \frac{1}{R}\right)}{\left[M\left(1 - \frac{1}{R}\right) - \Delta Q_m\right]} \tag{4.17c}$$

If data activity is actually constant and if the data actually behaves according to the model of Eq. (4.15), the queuing control system described by Eq. (4.17c) requires one measurement interval to calculate that tolerance which results in constant buffer queue. If data activity is not constant, or if the data do not behave according to the model of Eq. (4.15), buffer queue will asymptotically approach some stable value in all but the most pathological cases.

Let it be assumed that the estimated data activity is in error, so that $\hat{\alpha}_m$ differs from α_m by a small amount, ϵ_m. In other words,

$$\hat{\alpha}_m = \alpha_m + \epsilon_m \tag{4.18}$$

It can be seen, using Eq. (4.15), that

$$\Delta Q_m = M\left[1 - \frac{K_m}{(\hat{\alpha}_m - \epsilon_m)} - \frac{1}{R}\right] \tag{4.19a}$$

Another representation for ΔQ_m can be obtained by expanding the term $K_m/(\hat{\alpha}_m - \epsilon_m)$ from the above expression as follows

$$\frac{K_m}{(\hat{\alpha}_m - \epsilon_m)} = \left(\frac{K_m}{\hat{\alpha}_m}\right)\sum_{i=0}^{\infty}\left(\frac{\epsilon_m}{\hat{\alpha}_m}\right)^i$$

so that

$$\Delta Q_m = M\left(1 - \frac{K_m}{\hat{\alpha}_m} - \frac{1}{R}\right) - M\left(\frac{K_m}{\hat{\alpha}_m}\right)\left[\frac{\epsilon_m}{\hat{\alpha}_m} + \left(\frac{\epsilon_m}{\hat{\alpha}_m}\right)^2 + \dots\right] \tag{4.19b}$$

If the queuing control system is adjusted in accordance with Eq. (4.17c), the first bracketed term in Eq. (4.19b) is zero, so that this equation becomes

$$\Delta Q_m = -M\left(\frac{K_m}{\hat{\alpha}_m}\right)\left[\left(\frac{\epsilon_m}{\hat{\alpha}_m}\right) + \left(\frac{\epsilon_m}{\hat{\alpha}_m}\right)^2 + \ldots\right] \tag{4.19c}$$

If it is assumed that the error, ϵ_m, in estimation of data activity is independent of both $\hat{\alpha}_m$ and K_m, the result of averaging Eq. (4.19c) is

$$\overline{\Delta Q_m} = -M\left\{\overline{\left(\frac{K_m}{\hat{\alpha}_m^2}\right)}\,\overline{\epsilon_m} + \overline{\left(\frac{K_m}{\hat{\alpha}_m^3}\right)}\,\overline{\epsilon_m^2} + \ldots\right\} \tag{4.20a}$$

Now, if it is further assumed that the probability density of ϵ_m is an even function (equal numbers of positive and negative errors, roughly), the average value of all odd powers of ϵ_m is zero. Equation (4.20a) then reduces to

$$\overline{\Delta Q_m} = \Delta Q = -M\left\{\overline{\left(\frac{K_m}{\hat{\alpha}_m^3}\right)}\,\overline{\epsilon_m^2} + \overline{\left(\frac{K_m}{\hat{\alpha}_m^5}\right)}\,\overline{\epsilon_m^4} \ldots\right\} \tag{4.20b}$$

Because $\hat{\alpha}_m$ is greater than 1, K_m is less than 1, and ϵ_m is assumed small, Eq. (4.20b) converges to some finite value ΔQ. Since ΔQ is nonzero and negative, there is a tendency for a queuing control system of this sort, adjusted according to Eq. (4.17c), and behaving according to the stated assumptions, to empty the buffer. This tendency for the buffer to empty can be offset, however, by adjusting the system not according to Eq. (4.17c) but according to Eq. (4.17b) with $\hat{\Delta Q}_{m+1}$ set to some sufficiently positive value to offset the negative average buffer queue decrease, ΔQ, given by Eq. (4.20b). In other words, this system can be adjusted to consistently use a tolerance which is smaller than that value of tolerance which the model of Eq. (4.17b) predicts would be necessary for no resulting change in queue. If Δ is the projected increase in buffer queue which is adopted (i.e., $\hat{\Delta Q}_{m+1} = \Delta$) to offset the expected average decrease in buffer queue ΔQ, the tolerance K_{m+1}, which must be used during the $(m + 1)^{st}$ measurement interval, is given by Eq. (4.17b) as

$$\left(\frac{K_{m+1}}{K_m}\right) = \frac{\left\{M\left(1 - \frac{1}{R}\right) - \Delta\right\}}{\left\{M\left(1 - \frac{1}{R}\right) - \Delta Q_m\right\}} \tag{4.21}$$

Equation (4. 21) is plotted for several values of Δ in Fig. 4-12.

4. 3. 2 Preliminary Observations

Two tentative observations can be made from the preceding paragraphs. These are:

- Using predictive queuing control, data following a period of high data activity are not excessively degraded, as is the case with conventional queuing control techniques.
- Buffer queue stability is assured in the sense that an equilibrium value of buffer queue does exist for any given data history and activity.

Both of these observations are described in more detail in Par. 8. 1. 2.

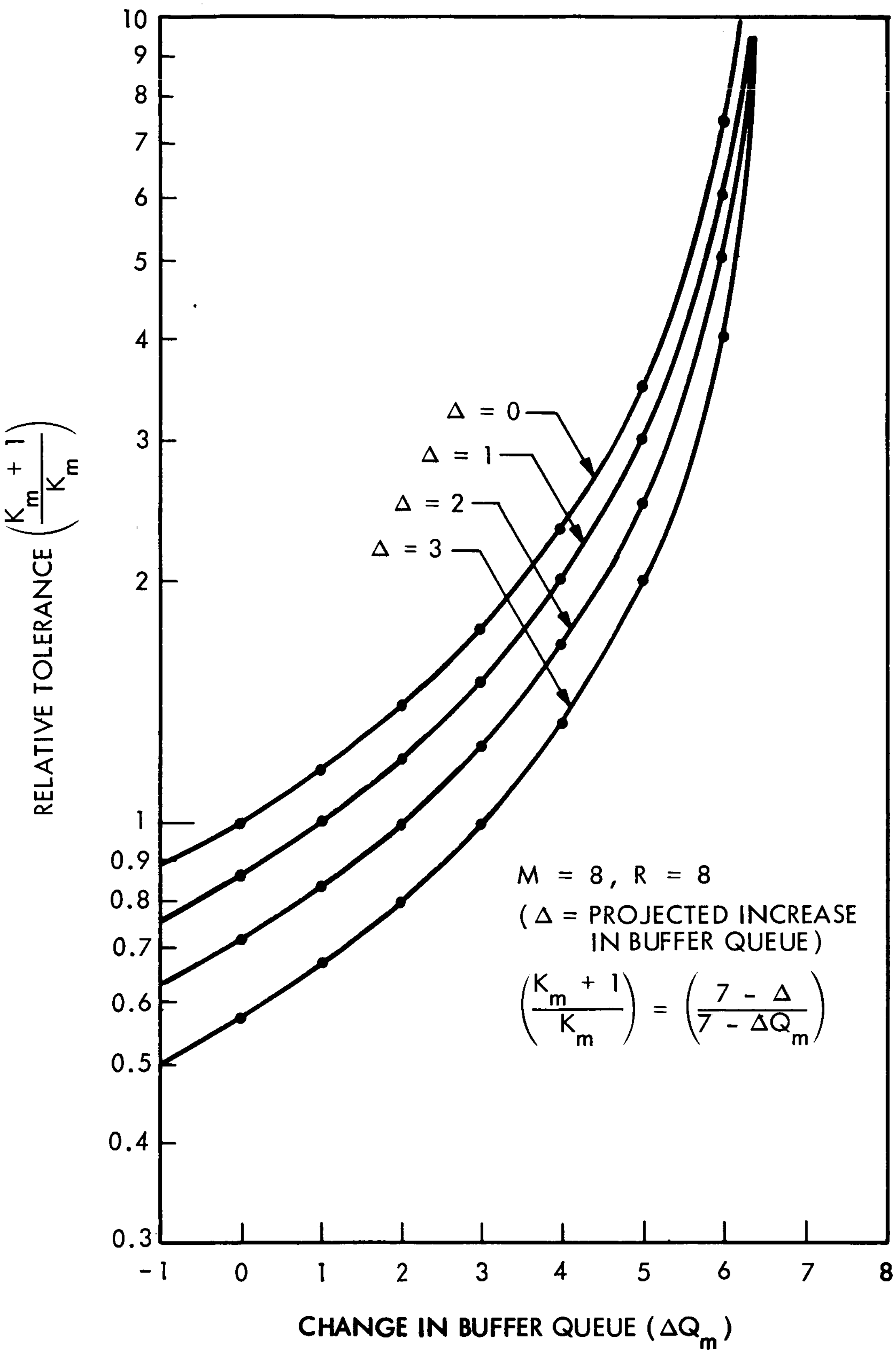


Fig. 4-12 Relative Tolerance K_{m+1}/K_m as a Function of Change in Queue ΔQ_m

4.4 OPTIMUM QUEUING CONTROL

This portion of the study was concerned with synthesizing an optimum queuing control function. The buffer model and optimality criterion which were adopted, as well as the methods used and the results obtained are described in the following paragraphs.

4.4.1 Buffer Model

The buffer model which was chosen is a single queue, single service buffer, with a first come, first serve queue discipline. It was assumed that a periodic attempt is made to read a word out of this buffer. Moreover, it was assumed that any number of words could be read into the buffer during the interval between readout attempts, this number of words being a random variable.

It was also assumed that measurement of buffer queue takes place immediately after each readout attempt, and that buffer size was infinite.

This model corresponds quite closely to a buffer which might be used in an actual data compression system. In the case of a real data compression system, however, there would be an upper bound on the number of words read into the buffer between readout attempts, imposed by the finite sample rate of the uncompressed data. In addition, the buffer in an actual data compression system would be of finite size. Because of the input data model chosen, however, these differences between an actual buffer and the model which was adopted are of little consequence.

4.4.2 Input Data Model

Work done on the initial phase of this contract has indicated that, when compressing multichannel Goddard S-6 Satellite data, the probability distribution of the number of words read into the buffer per unit time closely corresponds to an appropriate Poisson probability distribution. Figures 4-3 and 4-36 on pages 4.34 and 4.141 of Ref. 1 clearly show this Poisson behavior of buffer readins in these cases.